非线性滤波
理论与目标跟踪应用

Nonlinear Filtering Theory with Target Tracking Application

■ 占荣辉 张军 等著

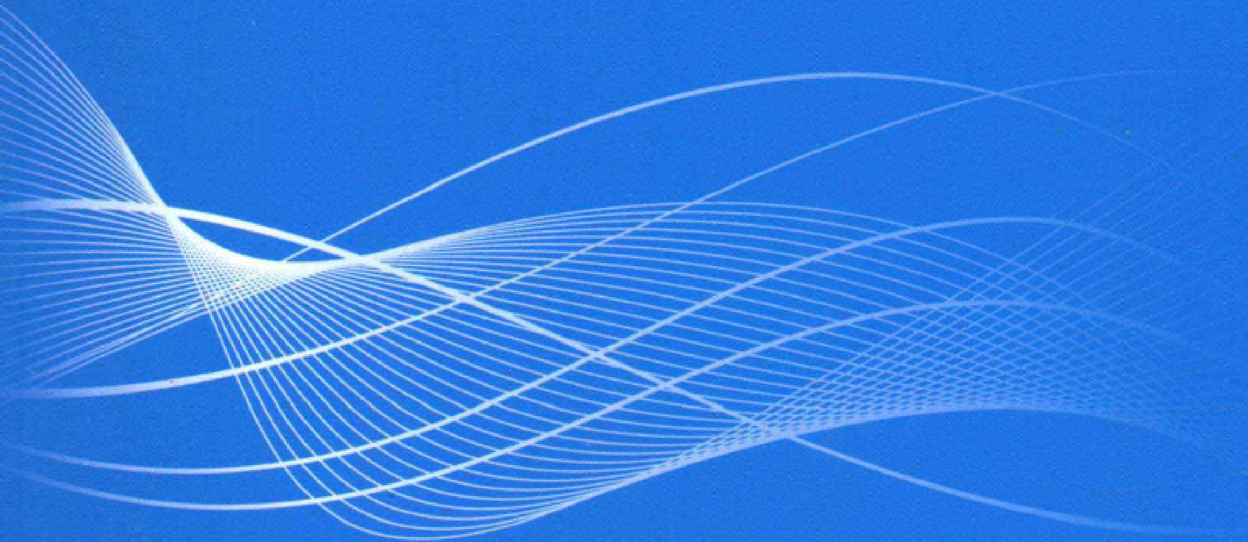


国防工业出版社
National Defense Industry Press

非线性滤波
理论与目标跟踪应用

Nonlinear Filtering Theory
with Target Tracking Application

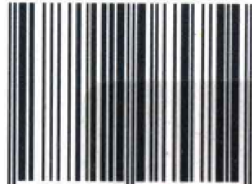




责任编辑：冯　晨　　cfeng@ndip.cn
责任校对：钱辉玲
封面设计：蒋秀芹



▶上架建议：信息处理/控制理论◀
http://www.ndip.cn
ISBN 978-7-118-08575-4



9 787118 085754 >
定价：59.90 元

非线性滤波理论与目标跟踪应用

占荣辉　张军　欧建平　胡杰民　著

国防工业出版社
·北京·

内容简介

本书以目标跟踪为应用背景，系统阐述了贝叶斯递推估计框架下的非线性滤波理论和实现方法，并针对不同的算法给出了大量的应用实例。全书内容共分四大部分、十二章，第一部分为非线性滤波理论与应用基础（分两章），主要包含卡尔曼滤波算法原理及各种衍生形式、目标跟踪的系统模型、性能测度等；第二部分为解析高斯近似滤波与应用（分四章），系统阐述四大类非线性滤波的算法原理、实现方法及改进算法，具体包括函数近似法（扩展卡尔曼滤波、中心差分卡尔曼滤波）、确定性采样近似法（不敏卡尔曼滤波）、积分近似法（求积卡尔曼滤波、求容积卡尔曼滤波）及混合近似法（高斯和扩展卡尔曼滤波、高斯和不敏卡尔曼滤波）等；第三部分为随机采样近似滤波与应用（分四章），详细阐述四类重要的随机采样近似滤波算法原理和实现方法，主要包括标准粒子滤波算法、序贯重要采样滤波（扩展粒子滤波、不敏粒子滤波）及改进算法、边缘化粒子滤波算法及智能优化采样滤波（包括遗传粒子滤波、免疫粒子滤波、人工鱼群粒子滤波）等；第四部分为非线性滤波的扩展与应用（分两章），重点阐述在多传感器和多模型两种扩展条件下、基于解析高斯和随机采样两种近似的滤波算法原理和实现方法，包括多传感器融合滤波、多模型滤波等。

本书可作为从事非线性滤波、贝叶斯估计、随机信号处理、目标跟踪等科研工作人员的参考用书，也可作为电子技术、信息技术、自动化技术等专业的研究生用书和补充教材。

图书在版编目（CIP）数据

非线性滤波理论与目标跟踪应用/占荣辉等著. —北京：国防工业出版社，2013.8

ISBN 978-7-118-08575-4

Ⅰ.①非… Ⅱ.①占… Ⅲ.①非线性滤波 Ⅳ.①0211.64

中国版本图书馆 CIP 数据核字（2013）第 096648 号

※

国防工業出版社 出版发行

（北京市海淀区紫竹院南路 23 号 邮政编码 100048）

三河市吉祥印务有限公司印刷

新华书店经售

*

开本 787×1092 1/16 印张 13¾ 字数 302 千字

2013 年 8 月第 1 版第 1 次印刷 印数 1—2000 册 **定价 59.90 元**

国防书店：（010）88540777 发行邮购：（010）88540776

发行传真：（010）88540755 发行业务：（010）88540717

前　言

客观世界中绝大部分物理系统都是非线性的，这一特点使经典的线性滤波理论在解决很多实际问题时面临着巨大的挑战，从而催生了非线性滤波理论的研究。在各种现实需求的强劲牵引和计算机技术发展的有力推动下，过去的20年～30年中各种非线性滤波理论和方法相继涌现，并被广泛应用于雷达、通信、导航、控制、金融和统计等众多军事和民用领域。

非线性滤波问题的本质，是利用含有噪声的观测量实现对非线性随机系统状态的估计。解决这一估计问题通常有递推处理和批处理两种方式，由于递推处理方式既不需要存储完整的数据集，也不必重新处理之前已经处理过的数据，因此在实际应用中兼具多方面的优势，特别适合实时应用场合。

作为非线性递推滤波的典型代表，扩展卡尔曼滤波（EKF）方法因其运算简单、方便实现等优势在过去的几十年中广为使用。但是，一方面，由于经典EKF仅采用一阶线性化近似，在系统非线性程度较高的条件下估计性能较差；另一方面，经典EKF滤波在实现过程中需要计算非线性函数的雅各比矩阵（即要求非线性函数解析可微），且不适用于非高斯噪声条件，大大制约了其应用范围，EKF的这些固有缺点促使人们不断寻求新的非线性滤波方法。

本书以目标跟踪为应用背景，系统梳理了近20年来非线性滤波领域的重大理论和学术思想，全面阐述了各种非线性滤波方法的理论基础、实现方法及适用条件等，并大量融入了作者近年来在这方面的理论研究和实践成果。

全书以不同条件下的贝叶斯推演算式求解方法为主线，将内容为四个部分。第一部分为非线性滤波理论与应用基础，共包含两章，主要介绍非线性滤波及目标跟踪应用的一些基础问题；第二部分为解析高斯近似滤波与应用，共包含四章，系统阐述几种基于解析高斯近似（如函数近似法、确定采样近似法、积分近似法及混合近似法等）的非线性滤波算法原理和实现方法；第三部分为随机采样近似滤波与应用，共包含四章，详细阐述基于蒙特卡罗模拟技术的随机采样近似滤波算法原理和实现方法；第四部分为非线性滤波的扩展与应用，共包含两章，重点阐述非线性滤波理论在多传感器和多模型两种扩展条件下的算法原理和实现方法。

作为作者及其所在科研团队多年来在该领域的研究和实践成果总结，本书具有以下几个显著特点：

系统性。将书中介绍的多种非线性滤波方法统一纳入到贝叶斯估计理论框架下，并指出各自的区别和联系，尽量使读者做到不仅知其然，而且知其所以然。

新颖性。注重以经典理论的方法论为基础，阐述非线性滤波方法的发展过程及其最新动态，其中部分内容为作者最新研究成果。

详实性。书中给出了大量针对不同应用的计算实例，这些实例涵盖了当前目标跟踪领域很多新的热点，书中介绍的绝大多数方法都经过作者的编程验证，具有很好的可重复性。

针对性。尽管书中介绍的各种方法具有广泛用途，但作者主要以雷达和红外传感器观测为侧重点讨论不同条件下的跟踪应用问题，避免了在内容安排方面的泛而空现象。

本书的出版得到了国防科学技术大学研究计划（JC08-04-04）、武器装备预研基金（xxx0909KG0107）及国家自然科学基金（61002022）的资助。另外，在本书撰写过程中，参考了国内外相关领域专家学者的研究成果，在此一并表达诚挚的谢意。

当前，各种新理论、新方法的发展日新月异，作为具有广泛应用前景和巨大扩展潜力的非线性滤波理论也是如此。尽管作者试图将这套理论方法以最恰当的方式呈现给读者，但受现有条件和作者水平所限，书中难免有疏漏和欠妥之处，恳请读者及同仁批评指正。

作　者

2012 年 10 月

符 号 说 明

符号	说明
$x/\boldsymbol{x}/\boldsymbol{X}$	标量/矢量/矩阵
$0/\mathbf{0}$	零标量/矢量或矩阵
$\hat{\boldsymbol{x}}_k$ / $\hat{\boldsymbol{x}}_{k\|k-1}$	$\boldsymbol{x}_k$ 的估计/预测
$(\cdot)^{\mathrm{T}}/(\cdot)^{-1}$	转置/求逆
$E[\cdot]$	数学期望
$\boldsymbol{P}_x = \mathrm{cov}(x)$	自协方差阵
$\boldsymbol{P}_{xy} = \mathrm{cov}(x, y)$	互协方差阵
$\mathbb{R}^n$	n 维实矢量
n_x	$\boldsymbol{x}$ 的维数
$\mathcal{N}(x;\mu,\sigma^2)$	高斯分布
σ^2/σ	方差/标准差
$\mathcal{U}(a,b)$	均匀分布
$\mathrm{diag}(\cdot)$	对角矩阵
$l(\cdot)$	似然函数
$(\cdot\|\cdot)$	条件概率
$P\{\cdot\}/\Pr(\cdot)$	事件的概率
$<\cdot,\cdot>$	矢量内积
$f(\cdot)$	非线性状态方程
$h(\cdot)$	非线性观测方程
$\boldsymbol{w}_k$	过程噪声矢量
$\boldsymbol{Q}_k = \mathrm{cov}[\boldsymbol{w}_k]$	过程噪声协方差
$\boldsymbol{v}_k$	观测噪声矢量
$\boldsymbol{R}_k = \mathrm{cov}[\boldsymbol{v}_k]$	观测噪声协方差
$\boldsymbol{P}_k$	估计误差协方差
$\boldsymbol{y}_{1:k} = \{\boldsymbol{y}_i\}_{i=1}^{k}$	累积观测序列
$\delta(\cdot)$	狄拉克函数
$p(\cdot)$	概率密度函数
arg max/arg min	使最大/最小化
$w/\tilde{w}$	权重/归一化权重
$r/\varphi/\phi$	距离/方位角/方向角
$I(\cdot)$	变量或函数积分式

$\sqrt{\boldsymbol{A}}$	矩阵方根且 $\boldsymbol{A}=\sqrt{\boldsymbol{A}}(\sqrt{\boldsymbol{A}})^{\mathrm{T}}$
$[\boldsymbol{A}]_i$	矩阵 $\boldsymbol{A}$ 的 i 列
$[\boldsymbol{A}]_{i,j}$	矩阵 $\boldsymbol{A}$ 的第(i,j)个元素
det[]	矩阵行列式
$\lVert(\cdot)\rVert$	2-范数

缩 略 词

AFSPF	artificial fish swarm particle filtering	人工鱼群粒子滤波
AIPF	artificial immune particle filtering	人工免疫粒子滤波
CDF	cumulative distribution function	累积分布函数
CDKF	central differences Kalman filtering	中心差分卡尔曼滤波
CMKF	converted measurement Kalman filtering	去偏转换卡尔曼滤波
CKF	cubature Kalman filtering	求容积卡尔曼滤波
EKF	extended Kalman filtering	扩展卡尔曼滤波
EPF	extended particle filtering	扩展粒子滤波
FIM	Fisher information matrix	Fisher 信息矩阵
GHQ	Gauss-Hermite quadrature	高斯-厄米特求积
QKF	quadrature Kalman filtering	求积卡尔曼滤波
GM	Gaussian mixture	高斯混合
GPF	genetic particle filtering	遗传粒子滤波
GSPF	Gaussian sum particle filtering	高斯和粒子滤波
IDF	importance density function	重要密度函数
IEKF	iterated extended Kalman filtering	迭代扩展卡尔曼滤波
IMM	interactive multiple model	交互多模
IS	importance sampling	重要性采样
IUKF	iterated unscented Kalman filtering	迭代不敏卡尔曼滤波
JEUKF	joint estimate unscented Kalman filtering	联合估计不敏卡尔曼滤波
MAP	maximum a posterior	最大后验
MCMC	Markov chain Monte Carlo	马尔可夫链蒙特卡罗
MGEKF	modified gain extended Kalman filtering	修正增益扩展卡尔曼滤波
MMPF	multiple model particle filtering	多模粒子滤波
MMSE	minimum mean-square error	最小均方误差
MPF	marginalized particle filtering	边缘化粒子滤波
MSE	mean square error	均方差
MUPF	modified unscented particle filtering	修正不敏粒子滤波
MVEKF	modified covariance extended Kalman filtering	修正协方差扩展卡尔曼滤波
NEES	normalized estimation error squared	归一化估计方差
NMEE	normalized mean estimation error	归一化平均估计误差
OSPA	optimal sub-pattern assignment	最优子模式分配

PCRB	posterior Cramer-Rao bound	后验克拉美-罗限
PDF	probability density function	概率密度函数
PF	particle filtering	粒子滤波
PHD	probability hypothesis density	概率假设密度
QKF	quadrature Kalman filtering	求积卡尔曼滤波
RFS	random finite sets	随机有限集
RMSE	root mean square error	均方根误差
SMC	sequential Monte Carlo	序贯蒙特卡罗
SRUKF	square-root unscented Kalman filtering	平方根不敏卡尔曼滤波
TSE	Taylor series expansion	泰勒级数展开
TbD	track-before-detect	检测前跟踪
UT	unscented transformation	不敏变换
UKF	unscented Kalman filtering	不敏卡尔曼滤波
UPF	unscented particle filtering	不敏粒子滤波

目　　录

第一部分　非线性滤波理论与应用基础

第二部分　解析高斯近似滤波与应用

第三部分 随机采样近似滤波与应用

第四部分 非线性滤波的扩展与应用

第一部分　非线性滤波理论与应用基础

客观世界中绝大部分问题都需要用非线性系统来描述，由于这一问题的普遍性，使得人们对非线性系统的建模和估计研究随着时间推移不断深入和拓展[1-10]。非线性滤波的本质是基于当前（或累积）系统量测，实现对非线性系统当前状态的准确估计。这里的非线性系统包含两层含义：一是系统，包含系统状态随时间的演化过程和观测手段，通常用状态方程和观测方程来建模；二是非线性，即状态方程和量测方程至少有一个是非线性的。在本书所讨论的系统中，状态方程用差分方程来描述，观测方程用离散时间观测模型来表示，由于考虑了过程噪声和量测噪声的影响，因此两者都是统计模型。

对于随机系统的估计问题，贝叶斯推演方法提供了一套严格、统一的处理框架，该方法通过利用所有可能获取的信息（包括观测序列）来构建系统状态的后验概率密度函数（PDF）。由于这一 PDF 包含了所有期望得到的统计信息，因此可认为是相应得到了被估计问题的完整的解，理论上通过该 PDF 也能得到系统状态（在某种准则下）的最优估计及其估计结果的性能测度。

在很多情况下（如一些对实时性要求高的场合），要求在新的观测序列到达时能即时得到状态的估计。在这种条件下，需要采用递推滤波方法，这就意味着对所获取的数据采用的是序贯处理而非批处理方式。在这种处理方法中，不需要存储完整的数据集，也不必重新处理之前已经处理过的数据，因此在实际应用中兼具多方面的优势。递推滤波主要通过两个步骤来完成，即预测与更新：在预测阶段，利用状态方程实现系统状态由当前时刻到下一个时刻 PDF 的前向预测；在更新阶段，主要利用新近的观测量并结合贝叶斯定理来修正状态的预测 PDF 从而得到其后验 PDF。

本部分主要介绍非线性滤波及目标跟踪应用的一些基础问题，第 1 章首先给出了贝叶斯滤波的统一框架，在此基础上针对一类特殊的（线性、高斯）系统，重点阐述了最优卡尔曼滤波的原理及其扩展应用方法。第 2 章则主要针对雷达/红外目标跟踪应用背景，介绍了常用的系统模型、常用的滤波性能评估指标及理论误差下限的计算方法等，并给出了具体的应用实例。

第 1 章　离散时间最优贝叶斯滤波

贝叶斯滤波为随机系统的状态估计问题提供了一套具有数学上严格定义、形式上完整表达的统一处理框架，这一处理框架给出了状态估计问题最优解的通用表达式（概念解）。但在大多数情况下（尤其是在非线性条件下），都无法获得这种通用数学公式的解析解（或闭式解）。不过，对于几种特殊的系统，闭式解的形式不但存在，而且可以方便获得，其中线性、高斯系统就是这样一类特殊的重要系统。

本章首先针对一般的离散时间系统模型，给出了贝叶斯滤波的求解过程及解的通用表达式，重点针对线性、高斯的特殊条件，在最大后验（MAP）估计和最小均方误差（MMSE）估计两种最优化准则下详细推导了最优卡尔曼（Kalman）滤波算法。并结合实际应用中面临的一些常见问题，给出了卡尔曼滤波算法的几种重要扩展形式，包括具有外部控制输入、含未知统计特性噪声、带恒定系统参数及求根实现等形式。

1.1　贝叶斯滤波的统一框架

考虑下列方程所描述的离散时间非线性系统

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}) + \boldsymbol{w}_k \tag{1-1}$$

$$\boldsymbol{y}_k = h(\boldsymbol{x}_k) + \boldsymbol{v}_k \tag{1-2}$$

式中：$\boldsymbol{x}_k \in \mathbb{R}^{n_x}$ 为系统状态；$f(\cdot)$ 和 $h(\cdot)$ 为已知非线性函数；$\boldsymbol{y}_k \in \mathbb{R}^{n_y}$ 为量测；$\boldsymbol{w}_k$ 和 $\boldsymbol{v}_k$ 分别为过程噪声和观测噪声。

从贝叶斯估计的方法论来看，k 时刻状态量 $\boldsymbol{x}_k$ 的后验密度能够提供该时刻状态完整的统计描述。滤波问题就是在给定观测 $\boldsymbol{y}_{1:k} = \{\boldsymbol{y}_i\}_{i=1}^{i=k}$ 的条件下得到时变状态量 $\boldsymbol{x}_k$ 的估计，这要求建立后验状态分布 $p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k})$ 的概率密度函数（PDF）。假定初始 PDF 为 $p(\boldsymbol{x}_0) \triangleq p(\boldsymbol{x}_0 \mid \boldsymbol{y}_0)$，则理论上 $p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k})$ 可通过以下两步递推计算求得。

预测： 假定 $k-1$ 时刻状态的后验 PDF 为 $p(\boldsymbol{x}_{k-1} \mid \boldsymbol{y}_{1:k-1})$，则其一步前向预测的 PDF 可根据 Chapman-Kolmogorov 方程计算为

$$p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1}) = \int p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}) p(\boldsymbol{x}_{k-1} \mid \boldsymbol{y}_{1:k-1}) \mathrm{d}\boldsymbol{x}_{k-1} \tag{1-3}$$

式中隐含假定了 $p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}) = p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}, \boldsymbol{y}_{1:k-1})$，$p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1})$ 称为状态的转移密度函数，用于描述式（1-1）中的一阶马尔可夫过程。

更新： 在获得 $p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1})$ 的基础上，综合新近的观测 $\boldsymbol{y}_k$，可通过如下的贝叶斯规则计算 k 时刻状态的后验 PDF

$$\begin{aligned} p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k}) &= p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1}, \boldsymbol{y}_k) \\ &= \frac{p(\boldsymbol{y}_k \mid \boldsymbol{x}_k, \boldsymbol{y}_{1:k-1}) p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1})}{p(\boldsymbol{y}_k \mid \boldsymbol{y}_{1:k-1})} = \frac{p(\boldsymbol{y}_k \mid \boldsymbol{x}_k) p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1})}{p(\boldsymbol{y}_k \mid \boldsymbol{y}_{1:k-1})} \end{aligned} \tag{1-4}$$

式中：$p(\boldsymbol{y}_k \mid \boldsymbol{y}_{1:k-1}) = \int p(\boldsymbol{y}_k \mid \boldsymbol{x}_k) p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1}) \mathrm{d}\boldsymbol{x}_k$ 为归一化系数。

式（1-3）和式（1-4）描述了一个由 $k-1$ 时刻后验 PDF 向 k 时刻后验 PDF 递推的完整过程，从而构成了贝叶斯估计最优解的通用表示形式。通过后验分布 $p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k})$ 可得到不同准则条件下 $\boldsymbol{x}_k$ 的最优估计 $\hat{\boldsymbol{x}}_k$，如最小均方误差（MMSE）估计

$$\hat{\boldsymbol{x}}_k^{MMSE} \triangleq E[\boldsymbol{x}_k | \boldsymbol{y}_{1:k}] = \int \boldsymbol{x}_k p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k}) \mathrm{d}\boldsymbol{x}_k \tag{1-5}$$

和最大后验（MAP）估计

$$\hat{\boldsymbol{x}}_k^{MAP} \triangleq \arg\max_{\boldsymbol{x}_k} p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k}) \tag{1-6}$$

需要指出的是，式（1-3）和式（1-4）所描述的只是概念上的解的形式，在通常情况下往往难以得到闭式解。但在以下几种特殊的条件下[9,11]，贝叶斯估计的最优解是存在的：

（1）式（1-1）和式（1-2）所描述的是线性系统模型，且 $\boldsymbol{w}_k$ 和 $\boldsymbol{v}_k$ 为高斯噪声，此时卡尔曼滤波结果即为最优估计。

（2）状态空间为离散值，且其状态个数为固定有限值，此时网格化方法[7]为最优滤波算法。

（3）一类具有线性观测的特殊非线性系统模型[11]，此时也可得出最优解的形式。

1.2 最优卡尔曼滤波

1.2.1 卡尔曼滤波的 MAP 准则推导

在式（1-3）中，若 $p(\boldsymbol{x}_{k-1} \mid \boldsymbol{y}_{1:k-1})$ 满足高斯分布，同时有下列假设成立：

（1）$\boldsymbol{w}_k$ 和 $\boldsymbol{v}_k$ 为相互独立的零均值高斯噪声，其协方差分别为 $\boldsymbol{Q}_k$ 和 $\boldsymbol{R}_k$；

（2）$f(\boldsymbol{x}_{k-1})$ 为 $\boldsymbol{x}_{k-1}$ 的线性函数，$h(\boldsymbol{x}_k)$ 为 $\boldsymbol{x}_k$ 的线性函数。

此时式（1-1）和式（1-2）可写为

$$\boldsymbol{x}_k = \boldsymbol{F}\boldsymbol{x}_{k-1} + \boldsymbol{w}_k \tag{1-7}$$

$$\boldsymbol{y}_k = \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{v}_k \tag{1-8}$$

则根据式（1-4）可知

$$p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k}) = \frac{p(\boldsymbol{y}_k \mid \boldsymbol{x}_k) p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1})}{p(\boldsymbol{y}_k \mid \boldsymbol{y}_{1:k-1})} \propto p(\boldsymbol{y}_k \mid \boldsymbol{x}_k) p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1}) \tag{1-9}$$

对式（1-9）取对数，得到式（1-6）的等价算式为

$$\hat{\boldsymbol{x}}_k^{MAP} = \arg\min_{\boldsymbol{x}_k} \{-\ln[p(\boldsymbol{y}_k \mid \boldsymbol{x}_k) p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1})]\} \tag{1-10}$$

容易证明，$p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1})$ 和 $p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k})$ 可以通过解析递推求得，且满足高斯分布，即

$$p(\boldsymbol{x}_{k-1} \mid \boldsymbol{y}_{1:k-1}) = \mathcal{N}(\boldsymbol{x}_{k-1}; \hat{\boldsymbol{x}}_{k-1}, \boldsymbol{P}_{k-1}) \tag{1-11}$$

$$\begin{aligned} p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1}) &= \mathcal{N}(\boldsymbol{x}_k; \hat{\boldsymbol{x}}_{k|k-1}, \boldsymbol{P}_{k|k-1}) \\ &= \frac{1}{\sqrt{(2\pi)^{n_x} \det[\boldsymbol{P}_{k|k-1}]}} \exp\left[-(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1})^{\mathrm{T}} \boldsymbol{P}_{k|k-1}^{-1} (\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1})\right] \end{aligned} \tag{1-12}$$

其中

$$\begin{aligned}\hat{\boldsymbol{x}}_{k|k-1}&=\int\boldsymbol{x}_k p(\boldsymbol{x}_k\mid\boldsymbol{y}_{1:k-1})\mathrm{d}\boldsymbol{x}_k=\iint\left[\int\boldsymbol{x}_k p(\boldsymbol{x}_k\mid\boldsymbol{x}_{k-1})\mathrm{d}\boldsymbol{x}_k\right]p(\boldsymbol{x}_{k-1}\mid\boldsymbol{y}_{1:k-1})\mathrm{d}\boldsymbol{x}_{k-1}\\&=\int\boldsymbol{F}\boldsymbol{x}_{k-1}p(\boldsymbol{x}_{k-1}\mid\boldsymbol{y}_{1:k-1})\mathrm{d}\boldsymbol{x}_{k-1}=\boldsymbol{F}\hat{\boldsymbol{x}}_{k-1}\end{aligned}\tag{1-13}$$

$$\begin{aligned}\boldsymbol{P}_{k|k-1}&=E\{[\boldsymbol{F}\boldsymbol{x}_{k-1}+\boldsymbol{w}_k-\boldsymbol{F}\hat{\boldsymbol{x}}_{k-1}][\boldsymbol{F}\boldsymbol{x}_{k-1}+\boldsymbol{w}_k-\boldsymbol{F}\hat{\boldsymbol{x}}_{k-1}]^{\mathrm{T}}\}\\&=E\{[\boldsymbol{F}(\boldsymbol{x}_{k-1}-\hat{\boldsymbol{x}}_{k-1})+\boldsymbol{w}_k][\boldsymbol{F}(\boldsymbol{x}_{k-1}-\hat{\boldsymbol{x}}_{k-1})+\boldsymbol{w}_k]^{\mathrm{T}}\}\\&=\boldsymbol{F}E[(\boldsymbol{x}_{k-1}-\hat{\boldsymbol{x}}_{k-1})(\boldsymbol{x}_{k-1}-\hat{\boldsymbol{x}}_{k-1})^{\mathrm{T}}]\boldsymbol{F}^{\mathrm{T}}+E[(\boldsymbol{w}_k\boldsymbol{w}_k^{\mathrm{T}})\\&=\boldsymbol{F}\boldsymbol{P}_{k-1}\boldsymbol{F}^{\mathrm{T}}+\boldsymbol{Q}_k\end{aligned}\tag{1-14}$$

式中：$\boldsymbol{P}_{k-1}=\mathrm{cov}[\boldsymbol{x}_{k-1}]=E[(\boldsymbol{x}_{k-1}-\hat{\boldsymbol{x}}_{k-1})(\boldsymbol{x}_{k-1}-\hat{\boldsymbol{x}}_{k-1})^{\mathrm{T}}]$。

又因为

$$\begin{aligned}p(\boldsymbol{y}_k\mid\boldsymbol{x}_k)&=\mathcal{N}(\boldsymbol{y}_k;\boldsymbol{H}\boldsymbol{x}_k,\boldsymbol{R})\\&=\frac{1}{\sqrt{(2\pi)^{n_y}\det[\boldsymbol{R}]}}\exp\left[-(\boldsymbol{y}_k-\boldsymbol{H}\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{R}^{-1}(\boldsymbol{y}_k-\boldsymbol{H}\boldsymbol{x}_k)\right]\end{aligned}\tag{1-15}$$

于是满足式（1-10）的 $\boldsymbol{x}_k$ 等价于

$$\frac{\partial}{\partial\boldsymbol{x}_k}[(\boldsymbol{y}_k-\boldsymbol{H}\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{R}^{-1}(\boldsymbol{y}_k-\boldsymbol{H}\boldsymbol{x}_k)]+\frac{\partial}{\partial\boldsymbol{x}_k}[(\boldsymbol{x}_k-\hat{\boldsymbol{x}}_{k|k-1})^{\mathrm{T}}\boldsymbol{P}_{k|k-1}^{-1}(\boldsymbol{x}_k-\hat{\boldsymbol{x}}_{k|k-1})]=0\tag{1-16}$$

$$-\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}^{-1}(\boldsymbol{y}_k-\boldsymbol{H}\boldsymbol{x}_k)+\boldsymbol{P}_{k|k-1}^{-1}(\boldsymbol{x}_k-\hat{\boldsymbol{x}}_{k|k-1})=0\tag{1-17}$$

$$\begin{aligned}\boldsymbol{P}_{k|k-1}^{-1}(\boldsymbol{x}_k-\hat{\boldsymbol{x}}_{k/k-1})&=\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}^{-1}(\boldsymbol{y}_k-\boldsymbol{H}\boldsymbol{x}_k)\\&=\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}^{-1}(\boldsymbol{y}_k-\boldsymbol{H}\boldsymbol{x}_k+\boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1}-\boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1})\\&=\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}^{-1}(\boldsymbol{y}_k-\boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1})-\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{H}(\boldsymbol{x}_k-\hat{\boldsymbol{x}}_{k|k-1})\end{aligned}\tag{1-18}$$

$$(\boldsymbol{P}_{k|k-1}^{-1}+\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{H})(\boldsymbol{x}_k-\hat{\boldsymbol{x}}_{k|k-1})=\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}^{-1}(\boldsymbol{y}_k-\boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1})\tag{1-19}$$

由于矩阵 $\boldsymbol{P}_{k|k-1}$ 和 $\boldsymbol{R}$ 均为对称正定阵，故 $\boldsymbol{P}_{k|k-1}^{-1}+\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{H}$ 可逆，于是由式（1-19），得

$$\hat{\boldsymbol{x}}_k=\hat{\boldsymbol{x}}_{k|k-1}+(\boldsymbol{P}_{k|k-1}^{-1}+\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}^{-1}(\boldsymbol{y}_k-\boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1})\tag{1-20}$$

若定义

$$\boldsymbol{K}_k\triangleq(\boldsymbol{P}_{k|k-1}^{-1}+\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}^{-1}=\boldsymbol{P}^*\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}^{-1}\tag{1-21}$$

对式（1-21）进行适当的变形，得

$$\begin{aligned}\boldsymbol{K}_k&=\boldsymbol{P}^*\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}^{-1}(\boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R})(\boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R})^{-1}\\&=\boldsymbol{P}^*(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{R})(\boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R})^{-1}\\&=\boldsymbol{P}^*(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{H}+\boldsymbol{P}_{k|k-1}^{-1})\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R})^{-1}\\&=\boldsymbol{P}^*(\boldsymbol{P}^*)^{-1}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R})^{-1}\\&=\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R})^{-1}\end{aligned}\tag{1-22}$$

同理可得

$$\begin{aligned}\boldsymbol{P}_k&=\mathrm{cov}[\boldsymbol{x}_k]=E\{(\boldsymbol{x}_k-\hat{\boldsymbol{x}}_k)(\boldsymbol{x}_k-\hat{\boldsymbol{x}}_k)^{\mathrm{T}}\}\\&=E\{[\boldsymbol{x}_k-\hat{\boldsymbol{x}}_{k|k-1}-\boldsymbol{K}_k(\boldsymbol{y}-\boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1})][\boldsymbol{x}_k-\hat{\boldsymbol{x}}_{k|k-1}-\boldsymbol{K}_k(\boldsymbol{y}-\boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1})]^{\mathrm{T}}\}\\&=\boldsymbol{P}_{k|k-1}^{-1}-\boldsymbol{P}_{k|k-1}^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{K}_k^{\mathrm{T}}-\boldsymbol{K}_k\boldsymbol{H}\boldsymbol{P}_{k|k-1}^{-1}+\boldsymbol{K}_k(\boldsymbol{H}\boldsymbol{P}_{k|k-1}^{-1}\boldsymbol{H}^{T}+\boldsymbol{R})\boldsymbol{K}_k^{\mathrm{T}}\end{aligned}\tag{1-23}$$

将式（1-22）中的$\boldsymbol{K}_k$代入式（1-23），并采用矩阵求逆引理（见附录A）化简，得

$$\begin{aligned}\boldsymbol{P}_k &= \boldsymbol{P}_{k|k-1} - \boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R})^{-1}\boldsymbol{H}\boldsymbol{P}_{k|k-1} \\ &= (\boldsymbol{P}_{k|k-1}^{-1} + \boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{H})^{-1} = \boldsymbol{P}^*\end{aligned} \tag{1-24}$$

于是有

$$\boldsymbol{P}_k = \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k\boldsymbol{H}\boldsymbol{P}_{k|k-1} = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H})\boldsymbol{P}_{k|k-1} \tag{1-25}$$

$$p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k}) = \mathcal{N}(\boldsymbol{x}_k;\hat{\boldsymbol{x}}_k, \boldsymbol{P}_k) \tag{1-26}$$

由此可得到完整的卡尔曼滤波递推算式：

$$\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{F}\hat{\boldsymbol{x}}_{k-1} \tag{1-27}$$

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{F}\boldsymbol{P}_{k-1}\boldsymbol{F}^{\mathrm{T}} + \boldsymbol{Q}_k \tag{1-28}$$

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k[\boldsymbol{y}_k - \boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1}] \tag{1-29}$$

$$\boldsymbol{P}_k = \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k\boldsymbol{H}\boldsymbol{P}_{k|k-1} \tag{1-30}$$

$$\boldsymbol{S}_k = \boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R}_k \tag{1-31}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{S}_k^{-1} \tag{1-32}$$

式中：$\boldsymbol{K}_k$为卡尔曼增益；$\boldsymbol{S}_k$为新息（$\boldsymbol{v}_k$=$\boldsymbol{y}_k - \boldsymbol{H}_k\hat{\boldsymbol{x}}_{k|k-1}$）的协方差矩阵。只要给定初始条件$\hat{\boldsymbol{x}}_0 = E[\boldsymbol{x}_0]$及$\boldsymbol{P}_0 = \mathrm{cov}[\boldsymbol{x}_0]$，通过式（1-27）～式（1-30）进行递推计算，即可动态得到状态量$\boldsymbol{x}_k$的滤波估计。

值得一提的是，由于$\boldsymbol{S}_k=\mathrm{cov}[\boldsymbol{v}_k]=\boldsymbol{P}_{yy}$，$\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}}=\mathrm{cov}[\tilde{\boldsymbol{x}}_{k|k-1}\tilde{\boldsymbol{y}}_{k|k-1}]=\boldsymbol{P}_{xy}$，实际应用中可根据具体需要将卡尔曼增益表示成不同的形式，如

$$\boldsymbol{K}_k=\boldsymbol{P}_{xy}\boldsymbol{P}_{yy}^{-1} \tag{1-33}$$

由此可得式（1-30）的另一种更新形式为

$$\boldsymbol{P}_k = \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k\boldsymbol{P}_{yy}\boldsymbol{K}_k^{\mathrm{T}} \tag{1-34}$$

1.2.2 卡尔曼滤波的正交投影推导

鉴于卡尔曼算法原理的重要性，下面将从另一个角度——正交投影原理出发对其进行再推导。

1. 正交投影原理

设$\boldsymbol{x}$和$\boldsymbol{z}$分别为n_x维和n_z随机矢量，$\boldsymbol{A}$为$n_x \times n_z$维矩阵，$\boldsymbol{b}$为n_x维非随机矢量。若存在一个n_x维矢量$\hat{\boldsymbol{x}}(\boldsymbol{z})$满足如下几个条件：

（1）$\hat{\boldsymbol{x}}(\boldsymbol{z})$可由$\boldsymbol{z}$线性表示，即存在

$$\hat{\boldsymbol{x}}(\boldsymbol{z})=\boldsymbol{A}\boldsymbol{z}+\boldsymbol{b}$$

（2）无偏性，即

$$E[\hat{\boldsymbol{x}}(\boldsymbol{z})]=E[\boldsymbol{x}]$$

（3）正交性，即

$$E\{[\boldsymbol{x} - \hat{\boldsymbol{x}}(\boldsymbol{z})]\boldsymbol{z}^{\mathrm{T}}\}=\boldsymbol{0}$$

则称$\hat{\boldsymbol{x}}(\boldsymbol{z})$是$\boldsymbol{x}$在$\boldsymbol{z}$上的正交投影，并记$\hat{\boldsymbol{x}}(\boldsymbol{z})=\hat{E}[\boldsymbol{x}|\boldsymbol{z}]$。

若将$\boldsymbol{x}$看做被估计量，$\boldsymbol{z}$为观测量，则当$\boldsymbol{x}$是基于$\boldsymbol{z}$的线性最小均方误差估计时，

恰好具备正交投影的上述三个条件，因此投影是存在的；反过来可以证明，具备这三个条件的 $\hat{\boldsymbol{x}}(\boldsymbol{z})$ 也只能是线性最小均方误差估计，也就说明投影是唯一的。

根据上面的正交投影定义，可得到以下几个重要的性质：

（1）设 $\boldsymbol{x}$ 和 $\boldsymbol{z}$ 分别为 n_x 维和 n_z 随机矢量，则 $\boldsymbol{x}$ 在 $\boldsymbol{z}$ 上的投影，唯一地等于基于 $\boldsymbol{z}$ 的线性最小均方误差估计，即

$$\hat{E}[\boldsymbol{x}|\boldsymbol{z}]=E[\boldsymbol{x}]+\mathrm{cov}(\boldsymbol{x},\boldsymbol{z})[\mathrm{cov}(\boldsymbol{z})]^{-1}\{\boldsymbol{z}-E[\boldsymbol{z}]\}$$

简单证明如下：

由正交投影条件（1）和（2）可知

$$E[\hat{\boldsymbol{x}}(\boldsymbol{z})]=\boldsymbol{A}E[\boldsymbol{z}]+\boldsymbol{b}=E[\boldsymbol{x}]$$

于是有

$$\boldsymbol{b}=E[\boldsymbol{x}]-\boldsymbol{A}E[\boldsymbol{z}]$$

$$\hat{\boldsymbol{x}}(\boldsymbol{z})=\boldsymbol{A}\boldsymbol{z}+E[\boldsymbol{x}]-\boldsymbol{A}E[\boldsymbol{z}]=E[\boldsymbol{x}]+\boldsymbol{A}\{\boldsymbol{z}-E[\boldsymbol{z}]\}$$

再根据正文投影条件（3），得

$$E\{[(\boldsymbol{x}-E[\boldsymbol{x}])-\boldsymbol{A}(\boldsymbol{z}-E[\boldsymbol{z}])]\boldsymbol{z}^{\mathrm{T}}\}=E\{[(\boldsymbol{x}-E[\boldsymbol{x}])-\boldsymbol{A}(\boldsymbol{z}-E[\boldsymbol{z}])](\boldsymbol{z}-E[\boldsymbol{z}]+E[\boldsymbol{z}])^{\mathrm{T}}\}=\boldsymbol{0}$$

对上式进行求解，得

$$\boldsymbol{A}=\mathrm{cov}(\boldsymbol{x},\boldsymbol{z})[\mathrm{cov}(\boldsymbol{z})]^{-1}$$

（2）设 $\boldsymbol{x}$ 和 $\boldsymbol{z}$ 分别为 n_x 维和 n_z 维随机矢量， $\boldsymbol{B}$ 为 $p\times n_x$ 维确定性矩阵，则

$$\hat{E}(\boldsymbol{B}\boldsymbol{x}|\boldsymbol{z})=\boldsymbol{B}\hat{E}(\boldsymbol{x}|\boldsymbol{z})$$

（3）设 $\boldsymbol{x}$ 为 n_x 维随机矢量，$\boldsymbol{z}_{1:k}=[\boldsymbol{z}_{1:k-1}^{\mathrm{T}}\ \ \boldsymbol{z}_k^{\mathrm{T}}]^{\mathrm{T}}$，$\boldsymbol{z}_k$ 为 n_z 维随机矢量，$\boldsymbol{z}_{1:k-1}$ 为 $n_z\times(k-1)$ 维矢量，则

$$\hat{E}(\boldsymbol{x}|\boldsymbol{z}_{1:k})=\hat{E}(\boldsymbol{x}|\boldsymbol{z}_{1:k-1})+\hat{E}(\tilde{\boldsymbol{x}}|\tilde{\boldsymbol{z}}_k) \tag{1-35}$$

式中：$\tilde{\boldsymbol{x}}=\boldsymbol{x}-\hat{E}(\boldsymbol{x}|\boldsymbol{z}_{1:k-1})$；$\tilde{\boldsymbol{z}}=\boldsymbol{z}_k-\hat{E}(\boldsymbol{z}_k|\boldsymbol{z}_{1:k-1})$。

证明：由于正交投影的无偏性，因而有 $E[\tilde{\boldsymbol{x}}]=\boldsymbol{0}$，$E[\tilde{\boldsymbol{z}}]=\boldsymbol{0}$。为了证明式(1-35)，首先令

$$\boldsymbol{x}^{*}=E[\boldsymbol{x}|\boldsymbol{z}_{1:k-1}]+\hat{E}[\tilde{\boldsymbol{x}}|\tilde{\boldsymbol{z}}_k]$$

且根据性质（1），有

$$\boldsymbol{x}^{*}=E[\boldsymbol{x}|\boldsymbol{z}_{1:k-1}]+E[\tilde{\boldsymbol{x}}\tilde{\boldsymbol{z}}_k^{\mathrm{T}}]\{E[\tilde{\boldsymbol{z}}_k\tilde{\boldsymbol{z}}_k^{\mathrm{T}}]\}^{-1}\tilde{\boldsymbol{z}}_k \tag{1-36}$$

现在只需证明 $\boldsymbol{x}^{*}$ 是 $\boldsymbol{x}$ 在 $\boldsymbol{z}_{1:k-1}$ 上的正交投影，为此只需根据正交投影的定义，验证 $\boldsymbol{x}^{*}$ 满足正交投影的三个条件。

① $\boldsymbol{x}^{*}$ 可由 $\boldsymbol{z}_{1:k-1}$ 线性表示，为此可令

$$\hat{E}[\boldsymbol{x}|\boldsymbol{z}_{1:k-1}]=\boldsymbol{b}_1+\boldsymbol{A}_1\boldsymbol{z}_{1:k-1} \tag{1-37}$$

$$\hat{E}[\boldsymbol{z}_k|\boldsymbol{z}_{1:k-1}]=\boldsymbol{b}_2+\boldsymbol{A}_2\boldsymbol{z}_{1:k-1} \tag{1-38}$$

将式（1-37）和式（1-38）代入式（1-35），并令

$$\boldsymbol{C}=E[\tilde{\boldsymbol{x}}\tilde{\boldsymbol{z}}_k^{\mathrm{T}}]\{E[\tilde{\boldsymbol{z}}\tilde{\boldsymbol{z}}_k^{\mathrm{T}}]\}^{-1}$$

由此可得

$$\begin{aligned}\boldsymbol{x}^{*}&=\boldsymbol{b}_1+\boldsymbol{A}_1\boldsymbol{z}_{1:k-1}+\boldsymbol{C}[\boldsymbol{z}_k-\boldsymbol{b}_2-\boldsymbol{A}_2\boldsymbol{z}_{1:k-1}]\\&=(\boldsymbol{b}_1-\boldsymbol{C}\boldsymbol{b}_2)+(\boldsymbol{A}_1-\boldsymbol{C}\boldsymbol{A}_2)\boldsymbol{z}_{1:k-1}+\boldsymbol{C}\boldsymbol{z}_k\\&=\boldsymbol{b}+[\boldsymbol{A}_1-\boldsymbol{C}\boldsymbol{A}_2\ \ \boldsymbol{C}]\begin{bmatrix}\boldsymbol{z}_{1:k-1}\\ \boldsymbol{z}_k\end{bmatrix}=\boldsymbol{b}+\boldsymbol{A}\boldsymbol{z}_{1:k}\end{aligned}$$

式中

$$\boldsymbol{b}=\boldsymbol{b}_1-\boldsymbol{C}\boldsymbol{b}_2,\quad \boldsymbol{A}=[\boldsymbol{A}_1-\boldsymbol{C}\boldsymbol{A}_2\boldsymbol{C}]$$

上述分析表明 $\boldsymbol{x}^{*}$ 是 $\boldsymbol{z}_{1:k}$ 的线性函数，亦即 $\boldsymbol{x}^{*}$ 可由 $\boldsymbol{z}_{1:k}$ 线性表示。

② $\boldsymbol{x}^{*}$ 满足无偏性，即 $E[\boldsymbol{x}^{*}]=E[\boldsymbol{x}]$。

直接对式（1-36）取期望，易得

$$E[\boldsymbol{x}^{*}]=E\{\hat{E}[\boldsymbol{x}|\boldsymbol{z}_{1:k-1}]\}+E[\tilde{\boldsymbol{x}}|\tilde{\boldsymbol{z}}_k^{\mathrm{T}}]\{E[\tilde{\boldsymbol{z}}_k\tilde{\boldsymbol{z}}_k^{\mathrm{T}}]\}^{-1}E[\tilde{\boldsymbol{z}}_k]=E[\boldsymbol{x}]$$

③ $\boldsymbol{x}^{*}$ 满足正交性，即 $E[(\boldsymbol{x}-\boldsymbol{x}^{*})\boldsymbol{z}_k^{\mathrm{T}}]=\boldsymbol{0}$。

首先有

$$E\{\tilde{\boldsymbol{x}}[\hat{E}(\boldsymbol{z}_k|\boldsymbol{z}_{1:k-1})]^{\mathrm{T}}\}=\boldsymbol{0}$$

$$E\{\tilde{\boldsymbol{z}}_k[\hat{E}(\boldsymbol{z}_k|\boldsymbol{z}_{1:k-1})]^{\mathrm{T}}\}=\boldsymbol{0}$$

根据上述两式可得

$$E\{\tilde{\boldsymbol{x}}\boldsymbol{z}_k^{\mathrm{T}}\}=E\{\tilde{\boldsymbol{x}}[\tilde{\boldsymbol{z}}_k+\hat{E}(\boldsymbol{z}_k|\boldsymbol{z}_{1:k-1})]^{\mathrm{T}}\}=E\{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{z}}_k^{\mathrm{T}}\}$$

$$E\{\tilde{\boldsymbol{z}}_k\boldsymbol{z}_k^{\mathrm{T}}\}=E\{\tilde{\boldsymbol{z}}_k[\tilde{\boldsymbol{z}}_k+\hat{E}(\boldsymbol{z}_k|\boldsymbol{z}_{1:k-1})]^{\mathrm{T}}\}=E\{\tilde{\boldsymbol{z}}_k\tilde{\boldsymbol{z}}_k^{\mathrm{T}}\}$$

现在证明正交性：

$$\begin{aligned}E[(\boldsymbol{x}-\boldsymbol{x}^{*})\boldsymbol{z}_k^{\mathrm{T}}]&=E\{[\boldsymbol{x}-\hat{E}(\boldsymbol{x}-\boldsymbol{z}_{1:k-1})-E[\tilde{\boldsymbol{x}}\tilde{\boldsymbol{z}}_k^{\mathrm{T}}](E[\tilde{\boldsymbol{z}}_k\tilde{\boldsymbol{z}}_k^{\mathrm{T}}])^{-1}\tilde{\boldsymbol{z}}_k\}\boldsymbol{z}_k^{\mathrm{T}}\\&=E[\tilde{\boldsymbol{x}}\boldsymbol{z}_{1:k}^{\mathrm{T}}]-E[\tilde{\boldsymbol{x}}\tilde{\boldsymbol{z}}_k^{\mathrm{T}}](E[\tilde{\boldsymbol{z}}_k\tilde{\boldsymbol{z}}_k^{\mathrm{T}}])^{-1}E[\tilde{\boldsymbol{z}}_k\boldsymbol{z}_{1:k}^{\mathrm{T}}]\\&=[E(\tilde{\boldsymbol{x}}\tilde{\boldsymbol{z}}_{1:k}^{\mathrm{T}})\ E(\tilde{\boldsymbol{x}}\tilde{\boldsymbol{z}}_k^{\mathrm{T}})]-E(\tilde{\boldsymbol{x}}\tilde{\boldsymbol{z}}_k^{\mathrm{T}})\,(E[\tilde{\boldsymbol{z}}_k\tilde{\boldsymbol{z}}_k^{\mathrm{T}}])^{-1}[E(\tilde{\boldsymbol{z}}_k\tilde{\boldsymbol{z}}_{1:k-1}^{\mathrm{T}})\ E(\tilde{\boldsymbol{z}}_k\tilde{\boldsymbol{z}}_k^{\mathrm{T}})]\\&=[\boldsymbol{0}\ \ E(\tilde{\boldsymbol{x}}\tilde{\boldsymbol{z}}_k^{\mathrm{T}})]-[\boldsymbol{0}\ \ E(\tilde{\boldsymbol{x}}\tilde{\boldsymbol{z}}_k^{\mathrm{T}})(E[\tilde{\boldsymbol{z}}_k\tilde{\boldsymbol{z}}_k^{\mathrm{T}}])^{-1}E(\tilde{\boldsymbol{z}}_k\tilde{\boldsymbol{z}}_k^{\mathrm{T}})]=\boldsymbol{0}\end{aligned}$$

正交性得证。

上述证明过程表明，$\boldsymbol{x}^{*}$ 满足正交投影的三个条件，是 $\boldsymbol{x}$ 在 $\boldsymbol{z}_{1:k}$ 上的正交投影，可表示为

$$\boldsymbol{x}^{*}=\hat{E}[\boldsymbol{x}|\boldsymbol{z}_{1:k}]=\hat{E}[\boldsymbol{x}|\boldsymbol{z}_{1:k-1}]+\hat{E}[\tilde{\boldsymbol{x}}|\tilde{\boldsymbol{z}}_k]$$

也可表示为

$$\hat{E}[\boldsymbol{x}|\boldsymbol{z}_{1:k}]=\hat{E}[\boldsymbol{x}|\boldsymbol{z}_{1:k-1}]+E[\tilde{\boldsymbol{x}}\tilde{\boldsymbol{z}}_k^{\mathrm{T}}][E(\tilde{\boldsymbol{z}}_k\tilde{\boldsymbol{z}}_k^{\mathrm{T}})]^{-1}\tilde{\boldsymbol{z}}_k$$

2. 卡尔曼滤波方程

仍然采用式（1-7）和式（1-8）的系统模型，假设 n_x 维随机矢量 $\boldsymbol{x}_j$ 基于观测量 $\boldsymbol{z}_{1:k}=\{\boldsymbol{z}_1,\boldsymbol{z}_2,\cdots,\boldsymbol{z}_k\}$ 的线性最小均方误差估计是 $\boldsymbol{x}_j$ 在 $\boldsymbol{z}_{1:k}$ 上的投影，表示为

$$\hat{\boldsymbol{x}}_{j|k}=\hat{E}[\boldsymbol{x}_j|\boldsymbol{z}_{1:k}]$$

当 $j=k$ 时，可得滤波估计

$$\hat{\boldsymbol{x}}_k\triangleq\hat{\boldsymbol{x}}_{k|k}=\hat{E}[\boldsymbol{x}_k|\boldsymbol{z}_{1:k}]=\hat{E}[\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1},\boldsymbol{z}_k]$$

由正交投影性质（3），得

$$\hat{\boldsymbol{x}}_k=\hat{E}[\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}]+\hat{E}[\tilde{\boldsymbol{x}}_{k|k-1}\tilde{\boldsymbol{z}}_{k|k-1}^{\mathrm{T}}][E(\tilde{\boldsymbol{z}}_{k|k-1}\tilde{\boldsymbol{z}}_{k|k-1}^{\mathrm{T}})]^{-1}\tilde{\boldsymbol{z}}_{k|k-1} \tag{1-39}$$

式中

$$\tilde{\boldsymbol{x}}_{k|k-1}=\boldsymbol{x}_k-\hat{E}[\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}]$$

$$\tilde{\boldsymbol{z}}_{k|k-1}=\boldsymbol{z}_k-\hat{E}[\boldsymbol{z}_k|\boldsymbol{z}_{1:k-1}]$$

式（1-39）即为卡尔曼滤波方程，下面讨论该式中各项的具体计算表达式。

$$\hat{E}[\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}]=\hat{\boldsymbol{x}}_{k|k-1}=\hat{E}\{\boldsymbol{F}\boldsymbol{x}_{k-1}+\boldsymbol{w}_{k-1}|\boldsymbol{z}_{1:k-1}\}=\boldsymbol{F}\hat{\boldsymbol{x}}_{k-1} \tag{1-40}$$

因为 $E[\boldsymbol{w}_{k-1}\boldsymbol{z}_{1:k-1}^{\mathrm{T}}]=\boldsymbol{0}$，故 $\hat{E}[\boldsymbol{w}_{k-1}\boldsymbol{z}_{k-1}]=\boldsymbol{0}$。

$\boldsymbol{z}_k$ 的预测误差为

$$\begin{aligned}\tilde{\boldsymbol{z}}_{k|k-1}&=\boldsymbol{z}_k-\hat{E}[\boldsymbol{z}_k|\boldsymbol{z}_{1:k-1}]=\boldsymbol{z}_k-\hat{E}[\boldsymbol{H}\boldsymbol{x}_k+\boldsymbol{v}_k|\boldsymbol{z}_{1:k-1}]\\&=\boldsymbol{z}_k-\hat{E}[\boldsymbol{H}\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}]=\boldsymbol{z}_k-\boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1}\end{aligned} \tag{1-41}$$

式（1-41）又可表示为

$$\tilde{\boldsymbol{z}}_{k|k-1}=\boldsymbol{z}_k-\hat{E}[\boldsymbol{z}_k|\boldsymbol{z}_{1:k-1}]=\boldsymbol{H}\boldsymbol{x}_k+\boldsymbol{v}_k-\boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1}=\boldsymbol{H}\tilde{\boldsymbol{x}}_{k|k-1}+\boldsymbol{v}_k$$

根据上式即可求得

$$E[\tilde{\boldsymbol{z}}_{k|k-1}\tilde{\boldsymbol{z}}_{k|k-1}^{\mathrm{T}}]=\boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R}_k \tag{1-42}$$

$$E[\tilde{\boldsymbol{x}}_{k|k-1}\tilde{\boldsymbol{z}}_{k|k-1}^{\mathrm{T}}]=E[\tilde{\boldsymbol{x}}_{k|k-1}(\boldsymbol{H}\tilde{\boldsymbol{x}}_{k|k-1}+\boldsymbol{v}_k)^{\mathrm{T}}]=\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}} \tag{1-43}$$

将式（1-40）～式（1-43）代入式（1-39），得

$$\hat{\boldsymbol{x}}_k=\hat{\boldsymbol{x}}_{k|k-1}+\boldsymbol{K}_k(\boldsymbol{z}_k-\boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1}) \tag{1-44}$$

式中

$$\boldsymbol{K}_k=\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R}_k)^{-1} \tag{1-45}$$

滤波误差协方差矩阵的推导与 1.2.1 节相类似，在此不再赘述。

1.3 卡尔曼滤波的扩展

1.3.1 完整形式的卡尔曼滤波

考虑线性动态系统

$$\begin{cases}\boldsymbol{x}_k=\boldsymbol{F}\boldsymbol{x}_{k-1}+\boldsymbol{\Gamma}\boldsymbol{u}_{k-1}+\boldsymbol{G}\boldsymbol{w}_{k-1}\\ \boldsymbol{z}_k=\boldsymbol{H}\boldsymbol{x}_k+\boldsymbol{D}\boldsymbol{v}_k\end{cases} \tag{1-46}$$

式中：$\boldsymbol{x}_k$ 是 n_x 维状态矢量；$\boldsymbol{z}_k$ 是 n_z 维观测矢量；$\boldsymbol{u}_{k-1}$ 为控制输入矢量；$\boldsymbol{F}$、$\boldsymbol{\Gamma}$ 和 $\boldsymbol{H}$ 为已知矩阵；$\boldsymbol{w}_{k-1}$ 和 $\boldsymbol{v}_k$ 是相互独立的高斯白噪声且有

$$E[\boldsymbol{w}_k]=\bar{\boldsymbol{w}},\quad \mathrm{cov}[\boldsymbol{w}_k]=\boldsymbol{Q}\delta_{kj} \tag{1-47}$$

$$E[\boldsymbol{v}_k]=\bar{\boldsymbol{v}},\quad \mathrm{cov}[\boldsymbol{v}_k]=\boldsymbol{R}\delta_{kj} \tag{1-48}$$

假设噪声均值 $\bar{\boldsymbol{w}}$、$\bar{\boldsymbol{v}}$ 和协方差 $\boldsymbol{Q}$、$\boldsymbol{R}$ 均为已知量，则此时的卡尔曼滤波算法为

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k \boldsymbol{\upsilon}_k \tag{1-49}$$

$$\boldsymbol{\upsilon}_k = \boldsymbol{z}_k - \boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1} - \boldsymbol{D}\overline{\boldsymbol{v}} \tag{1-50}$$

$$\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{F}\hat{\boldsymbol{x}}_{k-1} + \boldsymbol{\Gamma}\boldsymbol{u}_{k-1} + \boldsymbol{G}\overline{\boldsymbol{w}} \tag{1-51}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}}[\boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{D}\boldsymbol{R}\boldsymbol{D}^{\mathrm{T}}]^{-1} \tag{1-52}$$

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{F}\boldsymbol{P}_{k-1}\boldsymbol{F}^{\mathrm{T}} + \boldsymbol{G}\boldsymbol{Q}\boldsymbol{G}^{\mathrm{T}} \tag{1-53}$$

$$\boldsymbol{P}_k = [\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}]\boldsymbol{P}_{k|k-1} \tag{1-54}$$

1.3.2 噪声未知时的滤波算法

当噪声统计特性 $\overline{\boldsymbol{w}}$ 、$\overline{\boldsymbol{v}}$ 、$\boldsymbol{Q}$ 、$\boldsymbol{R}$ 未知时，连同状态 $\boldsymbol{x}_0,\boldsymbol{x}_1,\cdots,\boldsymbol{x}_k$ 的最大后验估计，$\overline{\boldsymbol{w}}$ 、$\overline{\boldsymbol{v}}$ 、$\boldsymbol{Q}$ 、 $\boldsymbol{R}$ 与 $\boldsymbol{x}_{j|k}$ 可通过最大化如下条件概率密度求得

$$J^* = p[\boldsymbol{x}_{0:k},\overline{\boldsymbol{w}},\overline{\boldsymbol{v}},\boldsymbol{Q},\boldsymbol{R}|z_{1:k}] \tag{1-55}$$

式中：$\boldsymbol{x}_{0:k}$ 和 $z_{1:k}$ 分别表示从初始时刻到 k 时刻目标的状态量和观测量，其实质是通过所有时刻的观测对以上五个参数进行估计。因为噪声是高斯白噪声，所以该条件概率密度可以写成解析表达式，并通过求对数、求导运算，得到最优解的表达式为

$$\hat{\boldsymbol{w}}_k = \frac{1}{k}\sum_{j=1}^{k}[\hat{\boldsymbol{x}}_j - \boldsymbol{F}\hat{\boldsymbol{x}}_{j-1|k} - \boldsymbol{\Gamma}\boldsymbol{u}_{j-1}] \tag{1-56}$$

$$\hat{\boldsymbol{Q}}_k = \frac{1}{k}\sum_{j=1}^{k}[\hat{\boldsymbol{x}}_{j|k} - \boldsymbol{F}\hat{\boldsymbol{x}}_{j-1|k} - \boldsymbol{\Gamma}\boldsymbol{u}_{j-1} - \overline{\boldsymbol{w}}][\hat{\boldsymbol{x}}_{j|k} - \boldsymbol{F}\hat{\boldsymbol{x}}_{j-1|k} - \boldsymbol{\Gamma}\boldsymbol{u}_{j-1} - \overline{\boldsymbol{w}}]^{\mathrm{T}} \tag{1-57}$$

$$\hat{\boldsymbol{v}}_k = \frac{1}{k}\sum_{j=1}^{k}[\boldsymbol{z}_j - \boldsymbol{H}\hat{\boldsymbol{x}}_{j|k}] \tag{1-58}$$

$$\hat{\boldsymbol{R}}_k = \frac{1}{k}\sum_{j=1}^{k}[\boldsymbol{z}_j - \boldsymbol{H}\hat{\boldsymbol{x}}_{j|k} - \overline{\boldsymbol{v}}][\boldsymbol{z}_j - \boldsymbol{H}\hat{\boldsymbol{x}}_{j|k} - \overline{\boldsymbol{v}}]^{\mathrm{T}} \tag{1-59}$$

以上解是最优解，因为式中的估计通过 k 个时刻的值进行平滑处理得到，所以最大程度地利用了观测得到的信息。但是在实际的应用中，这也将意味着要对数据进行批处理，实时性不强。为此，在误差能容忍范围内，用滤波估计 $\hat{\boldsymbol{x}}_j$ 或预测估计 $\hat{\boldsymbol{x}}_{j|j-1}$ 代替计算复杂的平滑估计 $\hat{\boldsymbol{x}}_{j|k}$ ，可求得次优 MAP 估计，即

$$\hat{\boldsymbol{w}}_k = \frac{1}{k}\sum_{j=1}^{k}[\hat{\boldsymbol{x}}_j - \boldsymbol{F}\hat{\boldsymbol{x}}_{j-1} - \boldsymbol{\Gamma}\boldsymbol{u}_{j-1}] \tag{1-60}$$

$$\hat{\boldsymbol{Q}}_k = \frac{1}{k}\sum_{j=1}^{k}(\hat{\boldsymbol{x}}_j - \boldsymbol{F}\hat{\boldsymbol{x}}_{j-1} - \boldsymbol{\Gamma}\boldsymbol{u}_{j-1} - \overline{\boldsymbol{w}})(\hat{\boldsymbol{x}}_j - \boldsymbol{F}\hat{\boldsymbol{x}}_{j-1} - \boldsymbol{\Gamma}\boldsymbol{u}_{j-1} - \overline{\boldsymbol{w}})^{\mathrm{T}} \tag{1-61}$$

$$\hat{\boldsymbol{v}}_k = \frac{1}{k}\sum_{j=1}^{k}(\boldsymbol{z}_j - \boldsymbol{H}\hat{\boldsymbol{x}}_{j|j-1}) \tag{1-62}$$

$$\hat{\boldsymbol{R}}_k = \frac{1}{k}\sum_{j=1}^{k}(\boldsymbol{z}_j - \boldsymbol{H}\hat{\boldsymbol{x}}_{j|j-1} - \overline{\boldsymbol{v}})(\boldsymbol{z}_j - \boldsymbol{H}\hat{\boldsymbol{x}}_{j|j-1} - \overline{\boldsymbol{v}})^{\mathrm{T}} \tag{1-63}$$

可以证明，由式（1-60）～式（1-63）给出的次优估计 $\hat{\boldsymbol{w}}$ 和 $\hat{\boldsymbol{v}}$ 是无偏的，但是 $\hat{\boldsymbol{Q}}$ 和 $\hat{\boldsymbol{R}}$ 是有偏的，可以求得修正后 $\hat{\boldsymbol{Q}}$ 和 $\hat{\boldsymbol{R}}$ 的次优无偏 MAP 估计为

$$\hat{\boldsymbol{Q}}_k=\frac{1}{k}\sum_{j=1}^{k}[\boldsymbol{K}_j\boldsymbol{v}_j\boldsymbol{v}_j^{\mathrm{T}}\boldsymbol{K}_j^{\mathrm{T}}+\boldsymbol{P}_j-\boldsymbol{F}\boldsymbol{P}_{j-1}\boldsymbol{F}^{\mathrm{T}}] \tag{1-64}$$

$$\hat{\boldsymbol{R}}_k=\frac{1}{k}\sum_{j=1}^{k}[\boldsymbol{v}_j\boldsymbol{v}_j^{\mathrm{T}}-\boldsymbol{H}\boldsymbol{P}_{j|j-1}\boldsymbol{H}^{\mathrm{T}}] \tag{1-65}$$

以上结果虽然无偏，但是因为有累加的过程，计算依然不方便，进一步推导可得次优无偏 MAP 估计的递推形式为

$$\hat{\boldsymbol{w}}_{k+1}=\frac{1}{k+1}(k\hat{\boldsymbol{w}}_k+\hat{\boldsymbol{x}}_{k+1}-\boldsymbol{F}\hat{\boldsymbol{x}}_k-\boldsymbol{\Gamma}\boldsymbol{u}_k) \tag{1-66}$$

$$\hat{\boldsymbol{Q}}_{k+1}=\frac{1}{k+1}(k\hat{\boldsymbol{Q}}_k+\boldsymbol{K}_{k+1}\boldsymbol{v}_{k+1}\boldsymbol{v}_{k+1}^{\mathrm{T}}\boldsymbol{K}_{k+1}^{\mathrm{T}}+\boldsymbol{P}_{k+1}-\boldsymbol{F}\boldsymbol{P}_k\boldsymbol{F}^{\mathrm{T}}) \tag{1-67}$$

$$\hat{\boldsymbol{v}}_{k+1}=\frac{1}{k+1}(k\hat{\boldsymbol{v}}_k+\boldsymbol{z}_{k+1}-\boldsymbol{H}\hat{\boldsymbol{x}}_{k+1|k}) \tag{1-68}$$

$$\hat{\boldsymbol{R}}_{k+1}=\frac{1}{k+1}(k\hat{\boldsymbol{R}}_k+\boldsymbol{v}_{k+1}\boldsymbol{v}_{k+1}^{\mathrm{T}}+\boldsymbol{H}\boldsymbol{P}_{k+1|k}\boldsymbol{H}^{\mathrm{T}}) \tag{1-69}$$

通过以上公式，利用卡尔曼滤波算法，就可以即时估计出过程噪声和观测噪声的参数，实现自适应滤波。

1.3.3 带未知参量的滤波算法

在很多情况下，系统中还包含一未知的参数，与一般的系统状态（随机矢量）不同，未知量为确定量，此时式（1-1）和式（1-2）的一种重要扩展形式可写为

$$\boldsymbol{x}_k=\boldsymbol{F}(\boldsymbol{\theta})\boldsymbol{x}_{k-1}+\boldsymbol{w}_k \tag{1-70}$$

$$\boldsymbol{y}_k=\boldsymbol{H}(\boldsymbol{\theta})\boldsymbol{x}_k+\boldsymbol{v}_k \tag{1-71}$$

对于这类系统，在完成状态量 $\boldsymbol{x}_k$ 滤波估计的同时需要得到参数 $\boldsymbol{\theta}$ 的估计，该过程一般称为系统识别（或对应目标跟踪中的运动模式识别）。对上述问题的求解通常有两种方法，一种是将状态量和参量分开来考虑，分别用传统的卡尔曼滤波算法来处理（称为双重滤波，Dual Filtering），且两者交替进行，即状态估计的结果用于参量估计，而参量估计的结果反过来又为状态估计所用；另一种更常用的方法是将状态量和参量组成联合状态，再采用一般的卡尔曼滤波算法来处理（称为联合滤波，Joint Filtering）。下面主要针对第二种思路进行重点介绍。

将 $\boldsymbol{x}_k$ 与 $\boldsymbol{\theta}_k$ 进行组合得到扩展状态量 $\boldsymbol{x}_k^a=[\boldsymbol{x}_k^{\mathrm{T}}\ \boldsymbol{\theta}_k^{\mathrm{T}}]^{\mathrm{T}}$，并将确定参量 $\boldsymbol{\theta}_k$ 进行随机化

$$\boldsymbol{\theta}_k=\boldsymbol{\theta}_{k-1}+\varsigma_k \tag{1-72}$$

式中：ς_k 为零均值白高斯噪声，方差为 $\boldsymbol{S}_k$。值得注意的是，上述随机化过程是必需的，否则就无法得到 $\boldsymbol{\theta}_k$ 的估计结果。

在上述假设条件下，式（1-70）和式（1-71）可进一步写成

$$\boldsymbol{x}_k^a=\boldsymbol{F}^a\boldsymbol{x}_{k-1}^a+\boldsymbol{w}_k^a=\begin{bmatrix}\boldsymbol{F}(\boldsymbol{\theta}_{k-1}) & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I}_\theta\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_{k-1}\\ \boldsymbol{\theta}_{k-1}\end{bmatrix}+\begin{bmatrix}\boldsymbol{w}_k\\ \varsigma_k\end{bmatrix} \tag{1-73}$$

$$\boldsymbol{y}_k=\boldsymbol{H}^a\boldsymbol{x}_k^a+\boldsymbol{v}_k=\begin{bmatrix}\boldsymbol{H}(\boldsymbol{\theta}_k) & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_k\\ \boldsymbol{\theta}_k\end{bmatrix}+\boldsymbol{v}_k \tag{1-74}$$

由此可得到如下的滤波算法流程。

Alg（1.1）：Joint Estimation Kalman Filtering

步骤 1：初始化

$$\hat{\boldsymbol{x}}_0^a=\begin{bmatrix}\hat{\boldsymbol{x}}_0\\ \hat{\boldsymbol{\theta}}_0\end{bmatrix}=\begin{bmatrix}E(\boldsymbol{x}_0)\\ E(\boldsymbol{\theta}_0)\end{bmatrix},\quad \boldsymbol{P}_0=\begin{bmatrix}\mathrm{cov}(\boldsymbol{x}_0) & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{S}_0\end{bmatrix}$$

步骤 2：时间更新

$$\hat{\boldsymbol{x}}_{k|k-1}^a=\begin{bmatrix}\hat{\boldsymbol{x}}_{k|k-1}\\ \hat{\boldsymbol{\theta}}_{k|k-1}\end{bmatrix}=\begin{bmatrix}\boldsymbol{F}(\hat{\boldsymbol{\theta}}_{k-1}) & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I}_{\boldsymbol{\theta}}\end{bmatrix}\begin{bmatrix}\hat{\boldsymbol{x}}_{k-1}\\ \hat{\boldsymbol{\theta}}_{k-1}\end{bmatrix}$$

$$\boldsymbol{P}_{k|k-1}=\begin{bmatrix}\boldsymbol{F}(\hat{\boldsymbol{\theta}}_{k-1}) & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I}_{\boldsymbol{\theta}}\end{bmatrix}\boldsymbol{P}_{k-1}\begin{bmatrix}\boldsymbol{F}(\hat{\boldsymbol{\theta}}_{k-1}) & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I}_{\boldsymbol{\theta}}\end{bmatrix}^{\mathrm{T}}+\begin{bmatrix}\boldsymbol{Q}_k & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{S}_k\end{bmatrix}$$

步骤 3：量测更新

$$\hat{\boldsymbol{x}}_k^a=\begin{bmatrix}\hat{\boldsymbol{x}}_k\\ \hat{\boldsymbol{\theta}}_k\end{bmatrix}=\begin{bmatrix}\hat{\boldsymbol{x}}_{k|k-1}\\ \hat{\boldsymbol{\theta}}_{k|k-1}\end{bmatrix}+\boldsymbol{K}_k\left(\boldsymbol{y}_k-\left[\boldsymbol{H}_k(\hat{\boldsymbol{\theta}}_{k|k-1})\quad \boldsymbol{0}\right]\right)\hat{\boldsymbol{x}}_{k|k-1}^a$$

$$\boldsymbol{P}_k=\left(\boldsymbol{I}-\boldsymbol{K}_k[\boldsymbol{H}(\hat{\boldsymbol{\theta}}_{k|k-1})\quad \boldsymbol{0}]\right)\boldsymbol{P}_{k|k-1}$$

$$\boldsymbol{K}_k=\boldsymbol{P}_{k|k-1}[\boldsymbol{H}(\hat{\boldsymbol{\theta}}_{k|k-1})\quad \boldsymbol{0}]^{\mathrm{T}}\left([\boldsymbol{H}(\hat{\boldsymbol{\theta}}_{k|k-1})\quad \boldsymbol{0}]\boldsymbol{P}_{k|k-1}[\boldsymbol{H}(\hat{\boldsymbol{\theta}}_{k|k-1})\quad \boldsymbol{0}]^{\mathrm{T}}+\boldsymbol{R}_k\right)^{-1}$$

从上述计算过程可以看出，若将 $\boldsymbol{\theta}$ 建模为确定性常量，即 $\boldsymbol{\theta}_k=\boldsymbol{\theta}_{k-1}$， $\boldsymbol{S}_k=\boldsymbol{0}$ ，则经递推后得到 $\hat{\boldsymbol{\theta}}_{k|k-1}=\hat{\boldsymbol{\theta}}_{k-1}$，独立于量测数据，从而无法完成对 $\boldsymbol{\theta}$ 的量测修正和估计。

1.3.4 卡尔曼滤波的求根实现

对于由式（1-7）和式（1-8）所描述的线性、高斯系统，最优卡尔曼滤波算法中矩阵 $\boldsymbol{P}_k$ 的 Riccati 方程解要求该矩阵为对称的半正定方阵，但在实际应用中，受计算机有限字长等因素的影响，舍入误差的存在往往会导致 $\boldsymbol{P}_k$ 矩阵为非对称、非正定。对于状态矢量，若其所包含的某些元素在量级上相差悬殊，数值问题就尤为突出。

为了说明这一问题，假定 n 阶矩阵 $\boldsymbol{P}$ 具有 n 个特征值 $\sigma_i(i=1,2,\cdots,n)$ ，若所有特征值为正，则称 $\boldsymbol{P}$ 为可逆正定矩阵，其条件数定义为

$$\lambda(\boldsymbol{P})=\frac{\sigma_{\max}(\boldsymbol{P})}{\sigma_{\min}(\boldsymbol{P})} \tag{1-75}$$

当 $\lambda(\boldsymbol{P})\to\infty$ 则称矩阵为病态矩阵（条件数差）或奇异矩阵。

在卡尔曼滤波算法中，误差协方差矩阵 $\boldsymbol{P}=E[(\boldsymbol{x}-\hat{\boldsymbol{x}})(\boldsymbol{x}-\hat{\boldsymbol{x}})^{\mathrm{T}}]$，说明该矩阵为正定的或者等效说该矩阵所有的特征值均大于零。但若 $\boldsymbol{x}$ 的某些元素的估计精度要比其他元素高得多，例如

$$\boldsymbol{P}=\begin{bmatrix}10^{10} & 0\\ 0 & 10^{-10}\end{bmatrix} \tag{1-76}$$

式（1-76）说明，$\boldsymbol{x}[1]$ 的标准偏差为 10^5 ，而 $\boldsymbol{x}[2]$ 的标准偏差则为 10^{-5} ，这种现象在实际中确实存在，这是因为各个分量可能采用了不同的单位或者各个分量具有不同的可观测性。显然，根据前面的定义，此时矩阵的条件数 $\lambda(\boldsymbol{P})=20$ 。在这种情况下，计算机（处理器）很可能将其视为奇异矩阵。这是因为，假定运算过程中采用了十进制定点处理器，

10^{10}项在处理器中可正确表示，而10^{-10}项则被视为 0 来对待。由此可见，数学上，矩阵 $\boldsymbol{P}$ 是非奇异的，而在应用计算时，$\boldsymbol{P}$ 则是奇异的。

改善数值问题的一条有效途径是采用平方根滤波算法，对于矩阵 $\boldsymbol{P}$，若将其表示为

$$\boldsymbol{P}=\boldsymbol{S}\boldsymbol{S}^{\mathrm{T}}=\boldsymbol{S}^{\mathrm{T}}\boldsymbol{S}$$

则 $\boldsymbol{S}$ 称为 $\boldsymbol{P}$ 的平方根（Square Root）矩阵。值得注意的是上述定义式与 $\boldsymbol{P}=\boldsymbol{S}^2$ 是不同的，实际处理中由 $\boldsymbol{P}$ 得到 $\boldsymbol{S}$ 的解不是唯一的，最常用的求解方法之一是采用 Cholesky 分解。

通过求平方根处理，滤波算法中将采用矩阵 $\boldsymbol{S}$ 而非 $\boldsymbol{P}$ 进行递推。这样做的好处在于，根据式（1-75）的定义，此时

$$\frac{\sigma_{\max}(\boldsymbol{S})}{\sigma_{\min}(\boldsymbol{S})}=\sqrt{\frac{\sigma_{\max}(\boldsymbol{P})}{\sigma_{\min}(\boldsymbol{P})}}$$

即 $\lambda(\boldsymbol{S})=\sqrt{\lambda(\boldsymbol{P})}=10^5$，说明求根滤波能够提供 2 倍于标准算法的精度，可有效帮助避免数值计算引起的问题。

下面讨论平方根卡尔曼滤波的具体实现过程，在开始介绍之前，首先给出另一个重要的结论。设 $\boldsymbol{S}_0=[\mathrm{cov}(\boldsymbol{x}_0)]^{1/2}$，$\boldsymbol{L}_k$ 为下三角矩阵且 $\boldsymbol{L}_k\boldsymbol{L}_k^{\mathrm{T}}=\boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R}_k$，则 $\boldsymbol{S}_{k|k-1}$ 为矩阵

$$\boldsymbol{S}_{k|k-1}\boldsymbol{S}_{k|k-1}^{\mathrm{T}}=[\boldsymbol{F}\boldsymbol{S}_{k-1}\ \boldsymbol{Q}_{k-1}^{1/2}]_{n_x\times(n_x+p)}[\boldsymbol{F}\boldsymbol{S}_{k-1}\ \boldsymbol{Q}_{k-1}^{1/2}]^{\mathrm{T}} \tag{1-77}$$

的平方根，且

$$\boldsymbol{S}_k=\boldsymbol{S}_{k|k-1}[\boldsymbol{I}-\boldsymbol{S}_{k|k-1}^{\mathrm{T}}\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{L}_k^{\mathrm{T}})^{-1}(\boldsymbol{L}_k^{\mathrm{T}}+\boldsymbol{R}_k^{l})^{-1}\boldsymbol{H}\boldsymbol{S}_{k|k-1}] \tag{1-78}$$

式中：上标1/2 表示矩阵的任意平方根；$\boldsymbol{R}_k^l$ 为 $\boldsymbol{R}_k$ 求平方根得到的下三角矩阵；辅助矩阵 $\boldsymbol{S}_{k|k-1}$ 和 $\boldsymbol{S}_k$（$k=1,2,\cdots$）分别为协方差矩阵 $\boldsymbol{P}_{k|k-1}$ 和 $\boldsymbol{P}_k$ 的平方根矩阵，该矩阵既不一定是下三角阵也未必是正定阵。式（1-78）的详细证明见附录 B。

据此，可得到求根形式的卡尔曼滤波算法流程如下。

Alg（1.2）：Square Root Kalman Filtering

步骤 1：初始化

$$\boldsymbol{S}_0=[\mathrm{cov}(\boldsymbol{x}_0)]^{1/2},\quad \hat{\boldsymbol{x}}_0=E[\boldsymbol{x}_0]$$

步骤 2：for $k=1,2,\cdots$

计算平方根矩阵 $\boldsymbol{S}_{k|k-1}$、$\boldsymbol{L}_k$ 及 $\boldsymbol{S}_k$ 如下

$$\boldsymbol{S}_{k|k-1}=\sqrt{[\boldsymbol{F}\boldsymbol{S}_{k-1}\ \boldsymbol{Q}_{k-1}^{1/2}]_{n_x\times(n_x+p)}[\boldsymbol{F}\boldsymbol{S}_{k-1}\ \boldsymbol{Q}_{k-1}^{1/2}]^{\mathrm{T}}}$$

$$\boldsymbol{L}_k=\sqrt{\boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R}_k}$$

$$\boldsymbol{S}_k=\boldsymbol{S}_{k|k-1}[\boldsymbol{I}-\boldsymbol{S}_{k|k-1}^{\mathrm{T}}\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{L}_k^{\mathrm{T}})^{-1}(\boldsymbol{L}_k^{\mathrm{T}}+\boldsymbol{R}_k^{l})^{-1}\boldsymbol{H}\boldsymbol{S}_{k|k-1}]$$

步骤 3：for $k=1,2,\cdots$

得到状态量的滤波估计为

$$\hat{\boldsymbol{x}}_k=\boldsymbol{F}\hat{\boldsymbol{x}}_{k-1}+\boldsymbol{K}_k(\boldsymbol{y}_k-\boldsymbol{H}\boldsymbol{F}\hat{\boldsymbol{x}}_{k-1})$$

$$\boldsymbol{K}_k=\boldsymbol{S}_{k|k-1}\boldsymbol{S}_{k|k-1}^{\mathrm{T}}\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{L}_k^{\mathrm{T}})^{-1}\boldsymbol{L}_k^{-1}$$

从整个滤波算法流程看，求逆操作仅限于对三角矩阵进行（三角矩阵的求逆相对简单），且这些三角矩阵均为平方根矩阵。

1.4 应用实例

1.4.1 系统噪声自适应估计

设某线性系统的状态演化过程为

$$\boldsymbol{x}_k = \boldsymbol{F}\boldsymbol{x}_{k-1} + \boldsymbol{G}w_{k-1}$$

式中：$\boldsymbol{x}_k=[x_k\ \dot{x}_k]^{\mathrm{T}}$ 为 k 时刻系统状态；$\boldsymbol{G}$ 为噪声输入矩阵；w_{k-1} 为零均值高斯过程噪声，且

$$\boldsymbol{F}=\begin{bmatrix}1 & T\\ 0 & 1\end{bmatrix},\quad \boldsymbol{G}=\begin{bmatrix}T^2/2\\ T\end{bmatrix},\quad \mathrm{cov}[w_k]=q^2$$

观测方程满足

$$z_k = \boldsymbol{H}\boldsymbol{x}_k + v_k$$

式中：$\boldsymbol{H}=\begin{bmatrix}1 & 0\end{bmatrix}$ 为观测矩阵；v_k 为未知高斯观测噪声。

假设状态初值 $\boldsymbol{x}_0=[2000\ \ -10]^{\mathrm{T}}$，$T$=0.25，$q$=1，$E[v_k]$=0，$R=\mathrm{var}[v_k]=20^2$。进一步假定初始状态估计及协方差矩阵分别为 $\hat{\boldsymbol{x}}_0=[2030\ \ -15]^{\mathrm{T}}$，$\boldsymbol{P}_0=\mathrm{diag}([80^2\ \ 10^2])$；初始观测噪声均值及方差估计分别为 $\hat{v}_0=0$，$\hat{R}_0=25^2$，采用 1.3.2 节中介绍的噪声自适应卡尔曼滤波（AKF）算法对系统状态进行估计，所得结果如图 1.1～图 1.3 所示。

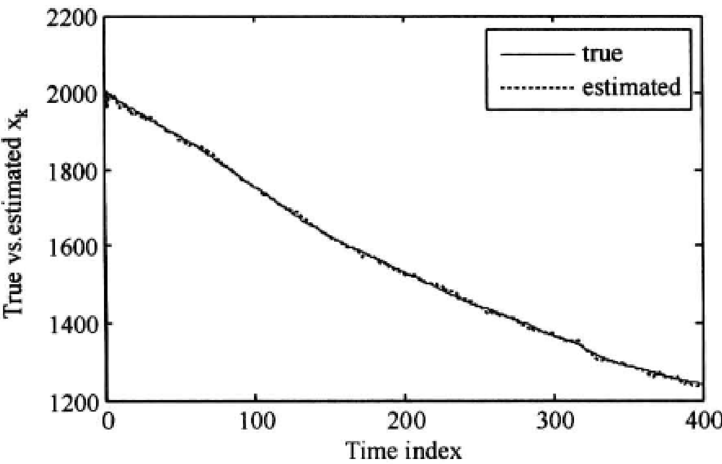


图 1.1 状态估计结果

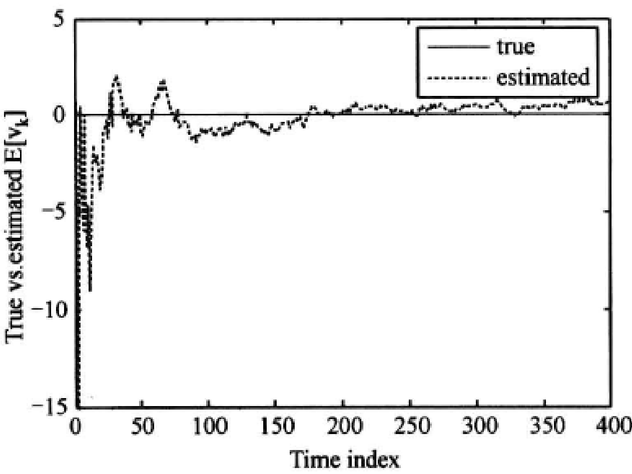


图 1.2 观测噪声均值估计

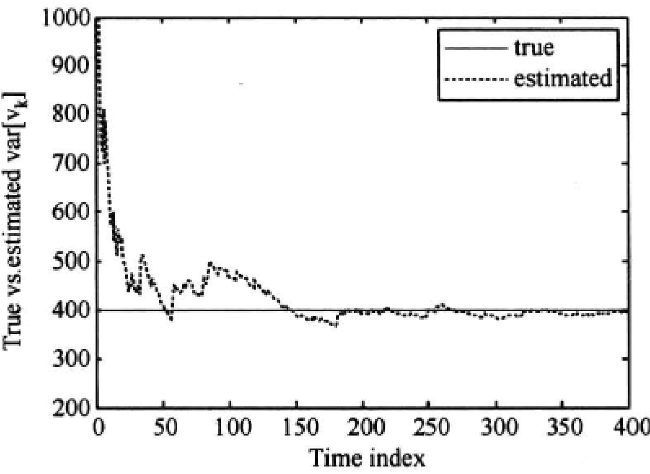


图 1.3 观测噪声方差估计

由图 1.1～图 1.3 中的结果可以看出，利用 AKF 可以得到系统状态和噪声特性参数（均值和方差）的估计，且随着递推过程的不断持续，估计误差逐渐减小并趋于稳定，说明了 AKF 算法的有效性。

1.4.2 匀加速机动目标跟踪

假设二维平面中的目标运动方程满足

$$\boldsymbol{x}_k = \boldsymbol{F}\boldsymbol{x}_{k-1} + \boldsymbol{\Gamma}\boldsymbol{u}_{k-1} + \boldsymbol{G}\boldsymbol{w}_{k-1}$$

式中：$\boldsymbol{x}_k=[x_k\ \dot{x}_k\ y_k\ \dot{y}_k]^{\mathrm{T}}$ 为 k 时刻目标的状态；$\boldsymbol{u}_k=[a_x\ a_y]^{\mathrm{T}}$ 为控制项；$\boldsymbol{F}$ 状态转移矩阵；$\boldsymbol{\Gamma}$ 为控制输入矩阵；$\boldsymbol{G}$ 为噪声输入矩阵；$\boldsymbol{w}_{k-1}$ 为零均值高斯过程噪声，且

$$\boldsymbol{F}=\begin{bmatrix}1 & T & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 1 & T\\0 & 0 & 0 & 1\end{bmatrix},\quad \boldsymbol{\Gamma}=\boldsymbol{G}=\begin{bmatrix}T^2/2 & 0\\T & 0\\0 & T^2/2\\0 & T\end{bmatrix},\quad \mathrm{cov}[\boldsymbol{w}_k]=\boldsymbol{Q}_k$$

观测方程为

$$\boldsymbol{z}_k = \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{v}_k$$

式中：$\boldsymbol{H}$ 为观测矩阵；$\boldsymbol{v}_k$ 为零均值高斯观测噪声，且

$$\boldsymbol{H}=\begin{bmatrix}1 & 0 & 0 & 0\\0 & 0 & 1 & 0\end{bmatrix},\quad \mathrm{cov}[\boldsymbol{v}_k]=\boldsymbol{R}_k$$

对跟踪系统而言，目标的状态及控制输入都是未知的，此时，可将目标状态和控制输入组成扩展状态量 $\boldsymbol{x}_k^a=[\boldsymbol{x}_k^{\mathrm{T}}\ \boldsymbol{u}_k^{\mathrm{T}}]^{\mathrm{T}}$，由此得到扩展条件下的状态方程和观测方程分别为

$$\boldsymbol{x}_k^a=\boldsymbol{F}^a\boldsymbol{x}_{k-1}^a+\boldsymbol{G}^a\boldsymbol{w}_k$$

$$\boldsymbol{y}_k^a = \boldsymbol{H}^a\boldsymbol{x}_k^a+\boldsymbol{v}_k$$

其中

$$\boldsymbol{F}^a=\begin{bmatrix}\boldsymbol{F} & \boldsymbol{\Gamma}\\ \boldsymbol{0}_{2\times4} & \boldsymbol{I}_2\end{bmatrix},\quad \boldsymbol{G}^a=\begin{bmatrix}\boldsymbol{G}\\ \boldsymbol{0}_{2\times2}\end{bmatrix},\quad \boldsymbol{H}^a=[\boldsymbol{HF}\ \ \boldsymbol{H\Gamma}]$$

假设目标初始状态为 $[20\mathrm{km}\ -200\mathrm{m/s}\ \ 15\mathrm{km}\ \ 100\mathrm{m/s}]^{\mathrm{T}}$，控制项为 $[6\ \ -8]^{\mathrm{T}}\mathrm{m/s^2}$，$T$=0.5s，过程噪声和观测噪声协方差为 $\boldsymbol{Q}_k=\mathrm{diag}([(2\mathrm{m/s^2})^2\ (2\mathrm{m/s^2})^2])$，$\boldsymbol{R}_k=\mathrm{diag}([(20\mathrm{m})^2\ (20\mathrm{m})^2])$，初始状态估计和协方差估计分别为

$$\hat{\boldsymbol{x}}_0^a=E[\boldsymbol{x}_0^a]=[20050\ \ -210\ \ 15050\ \ 110\ \ 0\ \ 0]^{\mathrm{T}}$$

$$\boldsymbol{P}_0^a=\mathrm{cov}[\boldsymbol{x}_0^a]=\mathrm{diag}([200^2\ 50^2\ 200^2\ 50^2\ 20^2\ 20^2])$$

利用联合估计卡尔曼滤波算法对目标扩展状态进行估计，所得结果如图 1.4～图 1.6 所示。

由图 1.4～图 1.6 中的结果看，利用状态量扩展方法并结合卡尔曼滤波，可实现对目标位置、速度及控制输入（恒定加速度）的有效估计，且估计误差在较短的时间内趋于收敛状态。另外，从图 1.7 和图 1.8 中不难看出，滤波器收敛后（如大于 10s），控制项的估值与真值曲线几乎完全重合，且估计误差在 $\pm1\sigma$ 范围之内（$\sigma_i=\left[\sqrt{\boldsymbol{P}_k^a}\right]_{i,i}$ (i=5,6)），说明结果是稳定的。

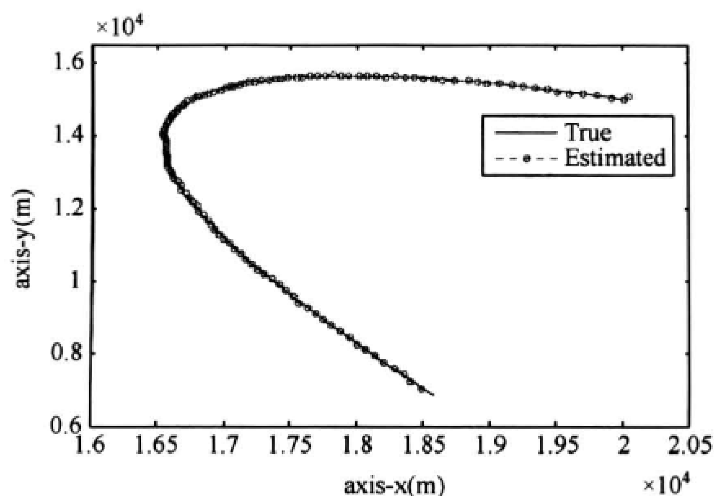


图 1.4 目标航迹跟踪

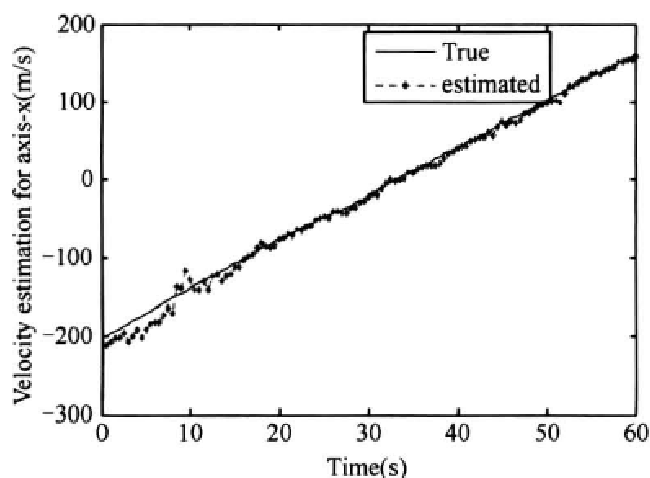


图 1.5 x 轴方向速度估计

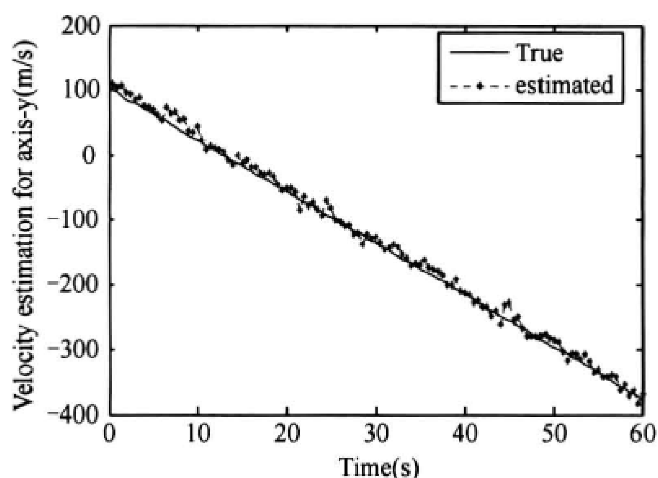


图 1.6 y 轴方向速度估计

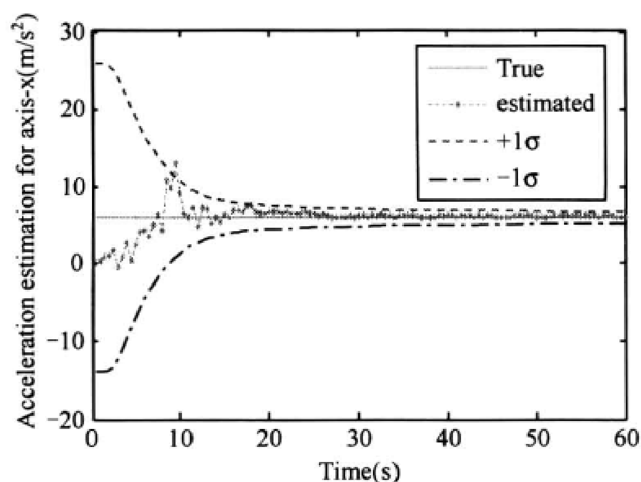


图 1.7 x 轴方向加速度估计

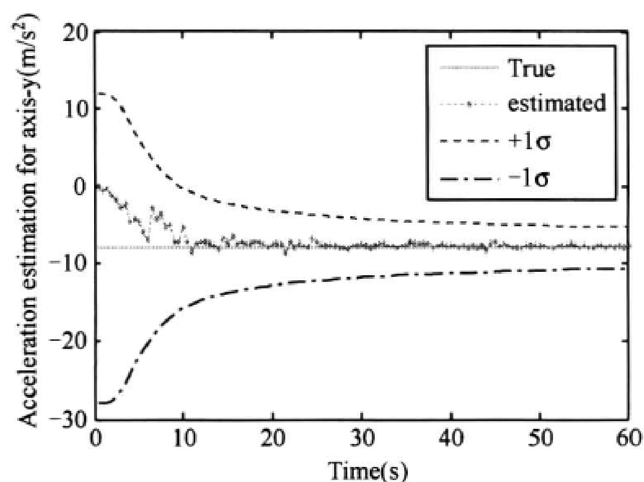


图 1.8 y 轴方向加速度估计

1.5 本章小结

本章主要介绍非线性滤波的原理基础。首先给出了贝叶斯滤波的求解通式，并针对线性、高斯的特殊条件，重点阐述了最优解的形式——卡尔曼滤波原理和实现方法。结合实际应用中面临的几类常见问题，本章还介绍了卡尔曼滤波算法的扩展形式，并给出

两种典型应用——噪声在线自适应估计和系统控制输入估计的卡尔曼滤波应用实例。

应该指出的是，卡尔曼滤波是在线性、高斯条件下最优贝叶斯滤波的特例之一，对于其他两种特例，由于其应用不具普遍性，本章没有作更多的探讨，有兴趣的读者可参阅相关的文献，如文献[9]、文献[11]等。另外，线性、高斯条件下的最优卡尔曼滤波原理是书中后续章节将详细阐述的各种（非线性、高斯/非高斯条件下）次优卡尔曼滤波原理和实现方法的基础，深入理解其推导过程和假设前提，对合理扩展和正确应用卡尔曼型滤波方法（在卡尔曼滤波框架下导出的滤波方法）具有重要的意义。

第 2 章　非线性跟踪应用与性能评估

由于客观世界中非线性问题的普遍性，使得非线性滤波理论和方法具有极为广泛的用途，其中最主要的应用领域之一是目标跟踪。

在非线性目标跟踪中，主要涉及三个重要问题，即模型、方法和评估。在这三个问题中，模型为基础，既包括目标的运动模型又包括观测模型，其中运动模型的选择取决于目标的类型及运动状态，而观测模型则由传感器类型及观测量决定；方法为核心，它是实现各种条件下目标状态滤波估计的重要手段；评估为补充，主要用于评价模型和方法的可行性、有效性和精确度等。

本章主要讨论模型和评估问题，重点针对雷达/红外目标跟踪应用，介绍常用的目标运动模型及在不同传感器条件下的观测模型。考虑到传统雷达观测的特点，专门介绍了二维和三维观测条件下的去偏转换方法。在此基础上，总结了目标跟踪中的一些常用评估指标。由于目标的跟踪滤波算法有很多，且各种算法的性能不一，在这种条件下，有必要寻求一种理论上最优的性能作为各种算法的评估参考标准。克拉美—罗限是不依赖于具体算法而能达到的状态估计误差理论下限，可用于评价各种算法能否达到最优及在何种程度上接近最优。有鉴于此，本章最后重点阐述了目标状态及混合状态两种条件下的克拉美—罗限的分析和计算问题。非线性滤波方法作为本书的核心部分将在后续的章节中分专题详细阐述。

2.1　非线性跟踪系统模型

2.1.1　目标运动常用模型

1．二维情况

1）近匀速运动

考虑直角坐标系中二维平面内的目标运动，若目标状态（包括位置和速度）为 $\boldsymbol{x}_k=[x\ \dot{x}\ y\ \dot{y}]^{\mathrm{T}}(k)$，则当目标做近匀速（CV）运动时，其状态方程可表示为

$$\boldsymbol{x}_k=\boldsymbol{F}_{\mathrm{CV}}\boldsymbol{x}_{k-1}+\boldsymbol{w}_k=\begin{bmatrix}\boldsymbol{F}_1 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{F}_1\end{bmatrix}\boldsymbol{x}_{k-1}+\boldsymbol{w}_k \tag{2-1}$$

$$\boldsymbol{Q}_k=\mathrm{cov}[\boldsymbol{w}_k]=q^2\cdot\begin{bmatrix}\boldsymbol{Q}_1 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{Q}_1\end{bmatrix} \tag{2-2}$$

式中

$$\boldsymbol{F}_1=\begin{bmatrix}1 & T\\ 0 & 1\end{bmatrix},\quad \boldsymbol{Q}_1=\begin{bmatrix}T^3/3 & T^2/2\\ T^2/2 & T\end{bmatrix}$$

T 为离散采样间隔。

2）近匀加速运动

若目标状态包括位置、速度和加速度，即 $\boldsymbol{x}_k=[x\ \dot{x}\ \ddot{x}\ y\ \dot{y}\ \ddot{y}]^{\mathrm{T}}(k)$，则近匀加速（CA）运动时的状态方程为

$$\boldsymbol{x}_k=\boldsymbol{F}_{\mathrm{CA}}\boldsymbol{x}_{k-1}+\boldsymbol{w}_k=\begin{bmatrix}\boldsymbol{F}_2 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{F}_2\end{bmatrix}\boldsymbol{x}_{k-1}+\boldsymbol{w}_k \tag{2-3}$$

$$\boldsymbol{Q}_k=\mathrm{cov}[\boldsymbol{w}_k]=q^2\cdot\begin{bmatrix}\boldsymbol{Q}_2 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{Q}_2\end{bmatrix} \tag{2-4}$$

式中

$$\boldsymbol{F}_2=\begin{bmatrix}1 & T & T^2\\ 0 & 1 & T\\ 0 & 0 & 1\end{bmatrix},\quad \boldsymbol{Q}_2=\begin{bmatrix}T^5/20 & T^4/8 & T^3/6\\ T^4/8 & T^3/3 & T^2/2\\ T^3/6 & T^2/2 & T\end{bmatrix}$$

3）近匀转弯运动

类似地，设 $\boldsymbol{x}_k=[x\ \dot{x}\ y\ \dot{y}]^{\mathrm{T}}(k)$，当目标做转弯率（角速度）为 ω 的近匀转弯（CT）机动时，其状态方程可表示为

$$\boldsymbol{x}_k=\boldsymbol{F}_{\mathrm{CT}}\boldsymbol{x}_{k-1}+\boldsymbol{w}_k=\begin{bmatrix}1 & \dfrac{\sin\omega T}{\omega} & 0 & \dfrac{\cos\omega T-1}{\omega}\\ 0 & \cos\omega T & 0 & -\sin\omega T\\ 0 & \dfrac{1-\cos\omega T}{\omega} & 1 & \dfrac{\sin\omega T}{\omega}\\ 0 & \sin\omega T & 0 & \cos\omega T\end{bmatrix}\boldsymbol{x}_{k-1}+\boldsymbol{w}_k \tag{2-5}$$

$$\begin{aligned}\boldsymbol{Q}_k&=\mathrm{cov}[\boldsymbol{w}_k]\\ &=q^2\begin{bmatrix}\dfrac{2(\omega T-\sin\omega T)}{\omega^3} & \dfrac{1-\cos\omega T}{\omega^2} & 0 & \dfrac{\omega T-\sin\omega T}{\omega^2}\\ \dfrac{1-\cos\omega T}{\omega^2} & T & -\dfrac{\omega T-\sin\omega T}{\omega^2} & 0\\ 0 & \dfrac{\omega T-\sin\omega T}{\omega^2} & \dfrac{2(\omega T-\sin\omega T)}{\omega^3} & \dfrac{1-\cos\omega T}{\omega^2}\\ \dfrac{\omega T-\sin\omega T}{\omega^2} & 0 & \dfrac{1-\cos\omega T}{\omega^2} & T\end{bmatrix}\end{aligned} \tag{2-6}$$

2. 三维情况

1）近匀速运动

考虑直角坐标系中三维空间中的目标运动，若目标状态（包括位置和速度）为 $\boldsymbol{x}_k=[x\ \dot{x}\ y\ \dot{y}\ z\ \dot{z}]^{\mathrm{T}}(k)$，则当目标做近匀速（CV）运动时，其状态方程可表示为

$$\boldsymbol{x}_k=\boldsymbol{F}_{\mathrm{CV}}\boldsymbol{x}_{k-1}+\boldsymbol{w}_k=\begin{bmatrix}\boldsymbol{F}_1 & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{F}_1 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{F}_1\end{bmatrix}\boldsymbol{x}_{k-1}+\boldsymbol{w}_k \tag{2-7}$$

$$\boldsymbol{Q}_k=\text{cov}[\boldsymbol{w}_k]=q^2\cdot\begin{bmatrix}\boldsymbol{Q}_1 & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{Q}_1 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{Q}_1\end{bmatrix} \tag{2-8}$$

式中

$$\boldsymbol{F}_1=\begin{bmatrix}1 & T\\ 0 & 1\end{bmatrix},\quad \boldsymbol{Q}_1=\begin{bmatrix}T^3/3 & T^2/2\\ T^2/2 & T\end{bmatrix}$$

2）近匀加速运动

若目标状态包括位置、速度和加速度，即 $\boldsymbol{x}_k=[x\ \dot{x}\ \ddot{x}\ y\ \dot{y}\ \ddot{y}\ z\ \dot{z}\ \ddot{z}]^{\mathrm{T}}(k)$，则近匀加速（CA）运动时的状态方程为

$$\boldsymbol{x}_k=\boldsymbol{F}_{\text{CA}}\boldsymbol{x}_{k-1}+\boldsymbol{w}_k=\begin{bmatrix}\boldsymbol{F}_2 & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{F}_2 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{F}_2\end{bmatrix}\boldsymbol{x}_{k-1}+\boldsymbol{w}_k \tag{2-9}$$

$$\text{cov}[\boldsymbol{w}_k]=q^2\cdot\begin{bmatrix}\boldsymbol{Q}_2 & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{Q}_2 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{Q}_2\end{bmatrix} \tag{2-10}$$

式中

$$\boldsymbol{F}_2=\begin{bmatrix}1 & T & T^2\\ 0 & 1 & T\\ 0 & 0 & 1\end{bmatrix},\quad \boldsymbol{Q}_2=\begin{bmatrix}T^5/20 & T^4/8 & T^3/6\\ T^4/8 & T^3/3 & T^2/2\\ T^3/6 & T^2/2 & T\end{bmatrix}$$

3）近匀转弯运动

三维条件下的匀转弯模型较为复杂，一种简化的途径是将目标运动约束于某一平面（水平面或斜平面）内，此时可得到斜平面中的近匀转弯模型。假定 $\boldsymbol{x}_k=[x\ \dot{x}\ \ddot{x}\ y\ \dot{y}\ \ddot{y}\ z\ \dot{z}\ \ddot{z}]^{\mathrm{T}}(k)$，转弯率的大小为 ω，则三维“空间中”的平面近匀转弯运动可表示为

$$\boldsymbol{x}_k=\boldsymbol{F}_{\text{CT}}\boldsymbol{x}_{k-1}+\boldsymbol{w}_k=\begin{bmatrix}\boldsymbol{F}_3 & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{F}_3 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{F}_3\end{bmatrix}\boldsymbol{x}_{k-1}+\boldsymbol{w}_k \tag{2-11}$$

$$\text{cov}[\boldsymbol{w}_k]=\begin{bmatrix}q_x^2\boldsymbol{Q}_3 & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & q_y^2\boldsymbol{Q}_3 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & q_z^2\boldsymbol{Q}_3\end{bmatrix} \tag{2-12}$$

式中

$$\boldsymbol{F}_3=\begin{bmatrix}1 & (\sin\omega T)/\omega & (1-\cos\omega T)/\omega^2\\ 0 & \cos\omega T & (\sin\omega T)/\omega\\ 0 & -\omega\sin\omega T & \cos\omega T\end{bmatrix}$$

$$
\boldsymbol{Q}_3=\begin{bmatrix} \dfrac{6\omega T-8\sin\omega T+\sin 2\omega T}{4\omega^5} & \dfrac{2\sin^4(\omega T/2)}{\omega^4} & \dfrac{-2\omega T+4\sin\omega T-\sin 2\omega T}{4\omega^3} \\ \dfrac{2\sin^4(\omega T/2)}{\omega^4} & \dfrac{2\omega T-\sin 2\omega T}{4\omega^3} & \dfrac{\sin^2\omega T}{2\omega^2} \\ \dfrac{-2\omega T+4\sin\omega T-\sin 2\omega T}{4\omega^3} & \dfrac{\sin^2\omega T}{2\omega^2} & \dfrac{2\omega T+\sin 2\omega T}{4\omega} \end{bmatrix}
$$

需要指出的是，上述给出的是几种最常用的模型，在目标具有复杂机动的条件下，其相应的模型也变得复杂。针对这一问题，感兴趣的读者可进一步参阅文献[12]、文献[13]等。

2.1.2 传感器观测模型

1. 点目标测量模型

如图 2.1 和图 2.2 所示为二维平面、三维空间条件下的目标（T）—传感器（O）观测几何关系，在点目标假设条件下，不同的传感器可得到不同的观测量，如典型的主动雷达观测可包括距离、方位/俯仰及径向速度等，被动雷达、红外传感器一般仅能提供方位/俯仰量测信息。下面分二维、三维两种情况介绍典型的点目标观测模型。

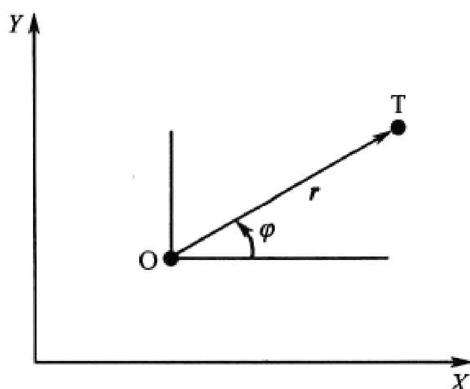


图 2.1 二维目标—传感器相对几何

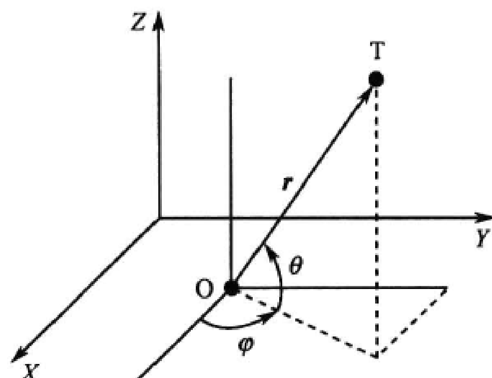


图 2.2 三维目标—传感器相对几何

1）二维情况

斜距：$r_k=h_r(\boldsymbol{x}_k)+v_{r,k}=\sqrt{x_k^2+y_k^2}+v_{r,k}$ （2-13）

方位：$\varphi_k=h_\varphi(\boldsymbol{x}_k)+v_{\varphi,k}=\arctan(y_k/x_k)+v_{\varphi,k}$ （2-14）

径向速度：$\dot{r}_k=h_{\dot{r}}(\boldsymbol{x}_k)+v_{\dot{r},k}=\dfrac{x_k\dot{x}_k+y_k\dot{y}_k}{\sqrt{x_k^2+y_k^2}}+v_{\dot{r},k}$ （2-15）

2）三维情况

斜距：$r_k=h_r(\boldsymbol{x}_k)+v_{r,k}=\sqrt{x_k^2+y_k^2+z_k^2}+v_{r,k}$ （2-16）

方位：$\varphi_k=h_\varphi(\boldsymbol{x}_k)+v_{\varphi,k}=\arctan(y_k/x_k)+v_{\varphi,k}$ （2-17）

俯仰：$\theta_k=h_\theta(\boldsymbol{x}_k)+v_{\theta,k}=\arctan(z_k/\sqrt{x_k^2+y_k^2})+v_{\theta,k}$ （2-18）

径向速度：$\dot{r}_k=h_{\dot{r}}(\boldsymbol{x}_k)+v_{\dot{r},k}=\dfrac{x_k\dot{x}_k+y_k\dot{y}_k+z_k\dot{z}_k}{\sqrt{x_k^2+y_k^2+z_k^2}}+v_{\dot{r},k}$ （2-19）

在上述模型中：$h(\cdot)$ 表示非线性观测函数；v 为观测噪声。

2. 点扩散模型

1）红外图像

对于非理想点目标，红外观测数据表现为信号强度在成像面中的扩散分布形式[14]。假设红外传感器产生的每一帧图像有 $n_x \times m_y$ 个分辨单元（像素），每一个分辨单元 (i,j) $(i=1,\cdots,n_x;j=1,\cdots,m_y)$ 对应一个矩形区域 $\Delta_x \times \Delta_y$，则 k 时刻传感器获取的 $n_x \times m_y$ 个强度观测数据可记为

$$z_k = \left\{ z_k^{(i,j)} : i=1,\cdots,n_x;j=1,\cdots,m_y \right\} \tag{2-20}$$

式中：$z_k^{(i,j)}$ 表示 k 时刻分辨单元 (i,j) 的强度观测，具体形式为

$$z_k^{(i,j)} = \begin{cases} h_k^{(i,j)}(\boldsymbol{x}_k) + v_k^{(i,j)} & H_1 \\ v_k^{(i,j)} & H_0 \end{cases} \tag{2-21}$$

式中：$h_k^{(i,j)}(\boldsymbol{x}_k) = \dfrac{\Delta_x \Delta_y I_k}{2\pi\Sigma^2} \exp\left[-\dfrac{(i\Delta_x - x_k)^2 + (j\Delta_y - y_k)^2}{2\Sigma^2} \right]$，$I_k$ 为信号强度；$v_k^{(i,j)}$ 为第 (i,j) 个分辨单元的测量噪声；H_1 和 H_0 代表目标信号的存在性假设，H_1 表示有目标，H_0 表示无目标（若无特殊说明，后文中均采用相同的表示方法）。

2）雷达图像

对雷达图像的建模可借鉴文献[15]、文献[16]中的方法，以回波信号的距离－多普勒－方位图作为雷达原始观测数据。限定感兴趣的传感器观测区域，以传感器所在位置为坐标原点，距离范围为 $[r_{\min}, r_{\max}]$，分为 N_r 个距离单元；多普勒速度范围为 $[d_{\min}, d_{\max}]$，分为 N_d 个多普勒单元；方位范围为 $[b_{\min}, b_{\max}]$，分为 N_b 个方位单元。雷达观测数据为散射能量，k 时刻一帧图像 z_k 包括 $N_r \times N_d \times N_b$ 个数据，记为

$$z_k = \left\{ z_k^{(i,j,l)} : i=1,\cdots,N_r;j=1,\cdots,N_d;l=1,\cdots,N_b \right\} \tag{2-22}$$

式中：$z_k^{(i,j,l)}$ 表示分辨单元 (i,j,l) 的能量观测，定义为

$$z_k^{(i,j,l)} = \left| z_{A,k}^{(i,j,l)} \right|^2 \tag{2-23}$$

式中：$z_{A,k}^{(i,j,l)}$ 表示目标的复幅度（复散射强度）观测数据，可表示为

$$z_{A,k}^{(i,j,l)} = \begin{cases} A_k h_A^{(i,j,l)}(\boldsymbol{x}_k) + v_k^{(i,j,l)} & H_1 \\ v_k^{(i,j,l)} & H_0 \end{cases} \tag{2-24}$$

式中：A_k 为目标的复幅度；$h_A^{(i,j,l)}(\boldsymbol{x}_k)$ 为分辨单元 (i,j,l) 的散射形式，可进一步定义为

$$A_k = \tilde{A}_k \mathrm{e}^{\mathrm{j}\phi_k}, \phi_k \in (0, 2\pi) \tag{2-25}$$

$$h_A^{(i,j,l)}(\boldsymbol{x}_k) = \exp\left\{ -\frac{(r_i - r_k)^2}{2r_c} l_r - \frac{(d_j - d_k)^2}{2d_c} l_d - \frac{(b_l - b_k)^2}{2b_c} l_b \right\} \tag{2-26}$$

式（2-26）可认为是雷达图像上目标信号强度分布的点扩散函数，r_c、d_c、b_c 为与距离、多普勒和方位单元尺寸有关的常数，分别由带宽、积累时间以及波束宽度决定。l_r、l_d 和 l_b 为损耗系数，r_k、d_k 和 b_k 分别为 k 时刻目标的真实距离、方位角和多普勒，其计算公式分别如式(2-13)～式(2-15)中的 $h_r(x_k)$、$h_\varphi(\boldsymbol{x}_k)$ 及 $h_{\dot{r}}(\boldsymbol{x}_k)$ 所示。$v_k^{(i,j,l)}$ 为分辨单元 (i,j,l) 的观测噪声，噪声在全空间内独立同分布，简记为 v_k，定义为复高斯形式，即

$$v_k = v_{I,k} + \mathrm{j}v_{Q,k} \tag{2-27}$$

式中：$v_{I,k}$ 和 $v_{Q,k}$ 均为 0 均值，方差为 σ^2 的高斯白噪声。

综上，可得分辨单元 (i,j,l) 的能量观测为

$$z_k^{(i,j,l)} = \left|z_{A,k}^{(i,j,l)}\right|^2 = \begin{cases} \left|A_k h_A^{(i,j,l)}(\boldsymbol{x}_k) + v_{I,k} + iv_{Q,k}\right|^2 & H_1 \\ \left|v_{I,k} + \mathrm{j}v_{Q,k}\right|^2 & H_0 \end{cases} \tag{2-28}$$

式（2-28）所描述的观测满足指数分布

$$p(z_k^{(i,j,l)} \mid \boldsymbol{x}_k) = \frac{1}{\mu_0^{(i,j,l)}} \exp\{-\frac{1}{\mu_0^{(i,j,l)}} z_k^{(i,j,l)}\} \tag{2-29}$$

其中参数 $\mu_0^{(i,j,l)}$ 的定义如下

$$\begin{aligned} \mu_0^{(i,j,l)} &= E_{v_{I,k},v_{Q,k}}[z_k^{(i,j,l)}] \\ &= \begin{cases} \tilde{A}_k^2 (h_A^{(i,j,l)}(\boldsymbol{x}_k))^2 + 2\sigma^2 & H_1 \\ 2\sigma^2 & H_0 \end{cases} \\ &= \begin{cases} P h_P^{(i,j,l)}(\boldsymbol{x}_k) + 2\sigma^2 \\ 2\sigma^2 \end{cases} \end{aligned} \tag{2-30}$$

式中：P 表示的回波信号强度；$h_P^{(i,j,l)}(\boldsymbol{x}_k)$ 定义为

$$\begin{aligned} h_P^{(i,j,l)}(\boldsymbol{x}_k) &= \left(h_A^{(i,j,l)}(\boldsymbol{x}_k)\right)^2 \\ &= \exp\left\{-\frac{(r_i - r_k)^2}{r_c} l_r - \frac{(d_j - d_k)^2}{d_c} l_d - \frac{(b_l - b_k)^2}{b_c} l_b\right\} \end{aligned} \tag{2-31}$$

3. 距离扩展模型

距离扩展模型主要是指在高分辨条件下，雷达目标强散射点在径向上的扩展，根据不同的目标外形，可分为以下几种扩展模型。

1）线形模型

线形模型[17,18]如图 2.3 所示，这种模型主要适用于长、宽比例极不协调的目标（即在某一维度上的扩展要远远大于另一维度）。设 ℓ 为目标的真实长度，则其在雷达视线 LOS（径向）上的长度可表示为

$$l_k = \ell\left|\cos\phi_k\right| + v_k = h(\boldsymbol{x}_k) + v_k = \ell\frac{\left|x_k\dot{x}_k + y_k\dot{y}_k\right|}{\sqrt{x_k^2 + y_k^2}\sqrt{\dot{x}_k^2 + \dot{y}_k^2}} + v_k \tag{2-32}$$

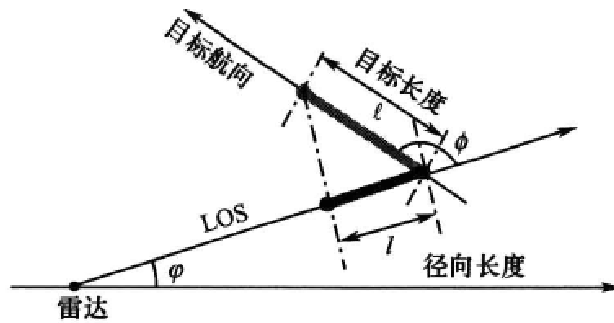


图 2.3　线形观测模型示意图

式中：ϕ_k 为传感器视线方向与目标速度方向的夹角（下同，该角度有时可等效为方向角[19]）；v_k 为径向长度观测误差。

2）椭圆模型

如图 2.4 所示，假设椭圆目标的长、短轴分别为 a 和 b，ℓ 为沿主轴方向的长度，则其在径向上的投影可表示为[20]

$$l_k=\ell\sqrt{\cos^2\phi_k+(b/a)^2\sin^2\phi_k}+v_k=h(\boldsymbol{x}_k)+v_k \tag{2-33}$$

式中：ϕ_k 的含义与式（2-32）中的相同，且

$$\phi_k=\arccos\frac{x_k\dot{x}_k+y_k\dot{y}_k}{\sqrt{x_k^2+y_k^2}\sqrt{\dot{x}_k^2+\dot{y}_k^2}} \tag{2-34}$$

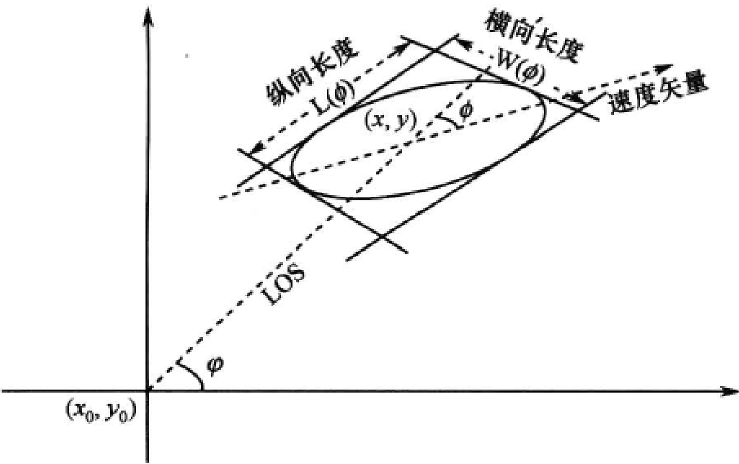


图 2.4 椭圆扩展模型示意图

3）矩形模型

如图 2.5 所示，假设目标呈方形，长度为 a、宽度为 b，其运动速度方向与雷达视线方向为 ϕ_k，则其在径向上的投影长度为[21]

$$l_k=a\cos\phi_k+b\sin\phi_k+v_k=h(\boldsymbol{x}_k)+v_k \tag{2-35}$$

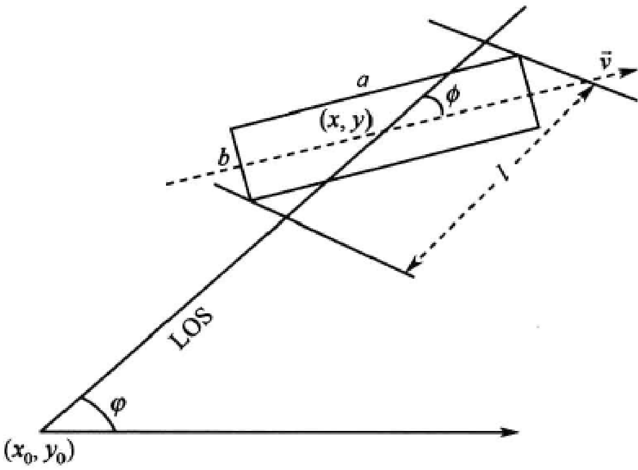


图 2.5 矩形扩展模型示意图

2.1.3 雷达量测转换

在雷达跟踪系统中，为方便描述，目标运动通常在直角坐标系中进行建模，而雷达量测一般却是在极坐标（二维）或球坐标（三维）系中得到的。这样，雷达目标跟踪就

是一个非线性问题，需要采用相应的非线性滤波算法来实现。在后续的章节中将会提到，在绝大多数情况下，非线性滤波算法都是次优算法。若能对非线性量测方程进行线性化转换，则能在最优线性卡尔曼滤波的框架下进行跟踪。去偏转换量测卡尔曼滤波（Converted Measurement Kalman Filtering，CMKF）[22,23]就是这样一类算法，先将雷达量测通过坐标变换表示为直角坐标系中量测的伪线性形式，然后估计转换测量误差的前两阶矩，最后通过线性卡尔曼滤波来处理。

1．二维去偏量测转换

在极坐标系中，相对于目标的真实斜距 r 和方位角 φ，雷达观测得到的斜距 r_m 和方位角 φ_m 可以定义为

$$\begin{cases} r_m = r + \tilde{r} \\ \varphi_m = \varphi + \tilde{\varphi} \end{cases} \tag{2-36}$$

假定斜距量测误差 $\tilde{r}$ 和方位角量测误差 $\tilde{\varphi}$ 为相互独立且均值为零的高斯噪声，方差分别为 σ_r^2 和 σ_θ^2，且有

$$\begin{cases} x_m = r_m \cos(\varphi_m) \\ y_m = r_m \sin(\varphi_m) \end{cases} \tag{2-37}$$

则式（2-37）可进一步写成

$$\begin{cases} x_m = (r+\tilde{r})\cos(\varphi+\tilde{\varphi}) = x+\tilde{x} \\ y_m = (r+\tilde{r})\sin(\varphi+\tilde{\varphi}) = y+\tilde{y} \end{cases} \tag{2-38}$$

将 $x = r\cos(\varphi)$、$y = r\sin(\varphi)$ 代入式（2-38）并适当变形，得

$$\begin{cases} \tilde{x} = r\cos\varphi(\cos\tilde{\varphi}-1) - \tilde{r}\sin\varphi\sin\tilde{\varphi} - r\sin\varphi\sin\tilde{\varphi} + \tilde{r}\cos\varphi\cos\tilde{\varphi} \\ \tilde{y} = r\sin\varphi(\cos\tilde{\varphi}-1) + \tilde{r}\cos\varphi\sin\tilde{\varphi} + r\cos\varphi\sin\tilde{\varphi} + \tilde{r}\sin\varphi\cos\tilde{\varphi} \end{cases} \tag{2-39}$$

由此可以看出，误差 $\tilde{x}$、$\tilde{y}$ 依赖于目标真实的距离、方位及其对应的观测误差，且彼此之间具有相关性。在量测噪声为零均值高斯噪声（即 $\tilde{\varphi} \sim \mathcal{N}(\tilde{\varphi};0,\sigma_\varphi^2)$）假设条件下，由于 $\sin\tilde{\varphi}$ 为奇函数，易得

$$E[\sin(\tilde{\varphi})] = 0$$

$$\begin{aligned} E[\cos\tilde{\varphi}] &= \int_{-\infty}^{+\infty} \cos\tilde{\varphi} \frac{1}{\sqrt{2\pi}\sigma_\varphi} \exp\left(-\frac{\tilde{\varphi}^2}{2\sigma_\varphi^2}\right) \mathrm{d}\tilde{\varphi} \\ &= \mathrm{Re}\left[\int_{-\infty}^{+\infty} (\cos\tilde{\varphi} + \mathrm{j}\sin\tilde{\varphi}) \frac{1}{\sqrt{2\pi}\sigma_\varphi} \exp\left(-\frac{\tilde{\varphi}^2}{2\sigma_\varphi^2}\right) \mathrm{d}\tilde{\varphi}\right] \\ &= \mathrm{Re}\left[\frac{1}{\sqrt{2\pi}\sigma_\varphi} \int_{-\infty}^{+\infty} \exp\left(\mathrm{j}\tilde{\varphi} - \frac{\tilde{\varphi}^2}{2\sigma_\varphi^2}\right) \mathrm{d}\tilde{\varphi}\right] \\ &= \mathrm{Re}\left\{\exp\left(-\frac{\sigma_\varphi^2}{2}\right) \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}\sigma_\varphi} \exp\left[-\left(\frac{\tilde{\varphi}}{\sqrt{2}\sigma_\varphi} - \mathrm{j}\frac{\sigma_\varphi}{\sqrt{2}}\right)^2\right] \mathrm{d}\tilde{\varphi}\right\} \end{aligned}$$

式中：$\mathrm{Re}(\cdot)$ 表示取复数的实部；j 为虚数单位（下同）。

令 $s = \frac{\tilde{\varphi}}{\sqrt{2}\sigma_\varphi} - \mathrm{j}\frac{\sigma_\varphi}{\sqrt{2}}$，代入上式，得

$$E[\cos\tilde{\varphi}]=\exp\left(-\frac{\sigma_\varphi^2}{2}\right)\int_{-\infty}^{+\infty}\frac{1}{\sqrt{\pi}}\exp(-s^2)\mathrm{d}s$$

$$=\exp\left(-\frac{\sigma_\varphi^2}{2}\right)=\mathrm{e}^{-\sigma_\varphi^2/2}$$

同理可得

$$E[\cos^2\tilde{\varphi}]=(1+\mathrm{e}^{-2\sigma_\varphi^2})/2$$

$$E[\sin^2\tilde{\varphi}]=(1-\mathrm{e}^{-2\sigma_\varphi^2})/2$$

$$E[\sin\tilde{\varphi}\cos\tilde{\varphi}]=0$$

令 $\boldsymbol{e}=[\tilde{x}\ \tilde{y}]^{\mathrm{T}}$，则有

$$\boldsymbol{\mu}=E[\boldsymbol{e}|r,\varphi]=[\mu_x\ \mu_y]^{\mathrm{T}}$$

$$\boldsymbol{R}=\mathrm{cov}[\boldsymbol{e}|r,\varphi]=\begin{bmatrix}\boldsymbol{R}_{xx} & \boldsymbol{R}_{xy}\\ \boldsymbol{R}_{yx} & \boldsymbol{R}_{yy}\end{bmatrix}$$

由此可得

$$\mu_x=r\cos\varphi(\mathrm{e}^{-\sigma_\varphi^2/2}-1) \tag{2-40}$$

$$\mu_y=r\sin\varphi(\mathrm{e}^{-\sigma_\varphi^2/2}-1) \tag{2-41}$$

$$\begin{aligned}\boldsymbol{R}_{xx}=\mathrm{var}[\tilde{x}|r,\varphi]=&r^2\mathrm{e}^{-\sigma_\varphi^2}[\cos^2\varphi(\cosh(\sigma_\varphi^2)-1)+\sin^2\varphi\sinh(\sigma_\varphi^2)]\\&+\sigma_r^2\mathrm{e}^{-\sigma_\varphi^2}[\cos^2\varphi\cosh(\sigma_\varphi^2)+\sin^2\varphi\sinh(\sigma_\varphi^2)]\end{aligned} \tag{2-42}$$

$$\begin{aligned}\boldsymbol{R}_{yy}=\mathrm{var}[\tilde{y}|r,\varphi]=&r^2\mathrm{e}^{-\sigma_\varphi^2}[\sin^2\varphi(\cosh(\sigma_\varphi^2)-1)+\cos^2\varphi\sinh(\sigma_\varphi^2)]\\&+\sigma_r^2\mathrm{e}^{-\sigma_\varphi^2}[\sin^2\varphi\cosh(\sigma_\varphi^2)+\cos^2\varphi\sinh(\sigma_\varphi^2)]\end{aligned} \tag{2-43}$$

$$\boldsymbol{R}_{yx}=\boldsymbol{R}_{xy}=\mathrm{cov}[\tilde{x},\tilde{y}|r,\varphi]=\sin\varphi\cos\varphi\mathrm{e}^{-2\sigma_\varphi^2}[\sigma_r^2+r^2(1-\mathrm{e}^{\sigma_\varphi^2})] \tag{2-44}$$

由式（2-40）～式（2-44）可以看出，量测转换的真实偏差及其协方差依赖于目标真实的距离和方位。但在实际应用中，目标的真实距离和方位信息是未知的。为此，在具体应用时，可在量测值 (r_m,φ_m) 已知的条件下对上述的真实均值和协方差矩阵求数学期望，得到平均真实偏差和平均真实协方差阵，即

$$\mu_{\mathrm{ave}}=\boldsymbol{E}[\mu|r_m,\varphi_m]=[\mu_{\mathrm{ave},x}\ \mu_{\mathrm{ave},y}]^{\mathrm{T}} \tag{2-45}$$

$$\boldsymbol{R}_{\mathrm{ave}}=E[\boldsymbol{R}|r_m,\varphi_m]=\begin{bmatrix}\boldsymbol{R}_{\mathrm{ave},xx} & \boldsymbol{R}_{\mathrm{ave},xy}\\ \boldsymbol{R}_{\mathrm{ave},yx} & \boldsymbol{R}_{\mathrm{ave},yy}\end{bmatrix} \tag{2-46}$$

综合利用式（2-36）、式（2-45）和式（2-46），得

$$\mu_{\mathrm{ave},x}=r_m\cos\varphi_m(\mathrm{e}^{-\sigma_\varphi^2}-\mathrm{e}^{-\sigma_\varphi^2/2})$$

$$\mu_{\mathrm{ave},y}=r_m\sin\varphi_m(\mathrm{e}^{-\sigma_\varphi^2}-\mathrm{e}^{-\sigma_\varphi^2/2})$$

$$\boldsymbol{R}_{\text{ave},xx}=r_m^2 \mathrm{e}^{-2\sigma_\varphi^2}\{\cos^2\varphi_m[\cosh(2\sigma_\varphi^2)-\cosh(\sigma_\varphi^2)]+\sin^2\varphi_m[\sinh(2\sigma_\varphi^2)-\sinh(\sigma_\varphi^2)]\}$$
$$+\sigma_r^2 \mathrm{e}^{-2\sigma_\varphi^2}\{\cos^2\varphi_m[2\cosh(2\sigma_\varphi^2)-\cosh(\sigma_\varphi^2)]+\sin^2\varphi_m[2\sinh(2\sigma_\varphi^2)-\sinh(\sigma_\varphi^2)]$$
$$\boldsymbol{R}_{\text{ave},yy}=r_m^2 \mathrm{e}^{-2\sigma_\varphi^2}\{\sin^2\varphi_m[\cosh(2\sigma_\varphi^2)-\cosh(\sigma_\varphi^2)]+\cos^2\varphi_m[\sinh(2\sigma_\varphi^2)-\sinh(\sigma_\varphi^2)]\}$$
$$+\sigma_r^2 \mathrm{e}^{-2\sigma_\varphi^2}\{\sin^2\varphi_m[2\cosh(2\sigma_\varphi^2)-\cosh(\sigma_\varphi^2)]+\cos^2\varphi_m[2\sinh(2\sigma_\varphi^2)-\sinh(\sigma_\varphi^2)]\}$$
$$\boldsymbol{R}_{\text{ave},yx}=\boldsymbol{R}_{\text{ave},xy}=\sin\varphi_m\cos\varphi_m \mathrm{e}^{-4\sigma_\varphi^2}[\sigma_r^2+(r_m^2+\sigma_r^2)(1-\mathrm{e}^{\sigma_\varphi^2})]$$

需要指出的是，按上述几式求得的平均协方差阵要大于真实协方差阵，这是由于以量测值为条件求数学期望引入了附加的误差。另外，只有距离较远、方位误差较大的情况下，位置误差和上面提到的协方差阵的增大才会显著。

由此可知，式（2-37）中的量测转换可修正为

$$\boldsymbol{z}_c=\begin{bmatrix} x_{c,m} \\ y_{c,m} \end{bmatrix}=\begin{bmatrix} r_m\cos\varphi_m \\ r_m\sin\varphi_m \end{bmatrix}-\begin{bmatrix} \mu_{\text{ave},x} \\ \mu_{\text{ave},y} \end{bmatrix}$$

此时，观测方程可建模为

$$\boldsymbol{z}_{c,k}=\boldsymbol{H}\boldsymbol{x}_k+\boldsymbol{v}_k$$

式中：$\boldsymbol{H}$ 为线性观测矩阵；$\boldsymbol{v}_k$ 为等效零均值量测噪声且 $E[\boldsymbol{v}_k\boldsymbol{v}_k^{\mathrm{T}}]\approx\boldsymbol{R}_{\text{ave},k}$。

为了验证上述去偏转换方法，假定目标真实距离和方位角分别为 r=10km，φ=45°，对应的观测精度为 σ_r=100m，σ_φ=1°。通过蒙特卡罗方法产生 500000 个样本并根据式（2-37）计算得到目标的位置均值 $(\hat{x},\hat{y})$ 与目标真实位置 (x,y) 的偏差及相应的协方差分别为 $(-1.067,-1.079)$ 和 $\begin{bmatrix} 20230 & -10229 \\ -10229 & 20230 \end{bmatrix}$，而通过式（2-40）～式（2-44）计算得到这一偏差值及协方差矩阵分别为 $(-1.0779,-1.077)$ 和 $\begin{bmatrix} 20229 & -10227 \\ -10227 & 20229 \end{bmatrix}$，由此可见，两种方法得到的结果非常接近，说明了去偏转换方法的正确性。

通过上述分析可知，对于距离—方位观测，虽然观测模型是非线性的，但经去偏转换后可变为线性观测模型，进而可以采用前文介绍的线性卡尔曼滤波算法进行处理。

2. 三维去偏量测转换

在球坐标中，相对于目标的真实距离 r、方位 φ 和俯仰 θ，雷达观测得到的斜距 r_m、方位角 φ_m 和俯仰角 θ_m 可表示为

$$\begin{cases} r_m=r+\tilde{r} \\ \varphi_m=\varphi+\tilde{\varphi} \\ \theta_m=\theta+\tilde{\theta} \end{cases} \tag{2-47}$$

式中，假定量测误差 $\tilde{r}$、$\tilde{\varphi}$ 及 $\tilde{\theta}$ 为零均值高斯噪声且相互独立，其方差分别为 σ_r^2、σ_φ^2 和 σ_θ^2。

采用与二维情况类似的方法，可得到球坐标系与直角坐标系下的转换关系为

$$\begin{cases} x_m=(r+\tilde{r})\cos(\varphi+\tilde{\varphi})\cos(\theta+\tilde{\theta})=x+\tilde{x} \\ y_m=(r+\tilde{r})\sin(\varphi+\tilde{\varphi})\cos(\theta+\tilde{\theta})=y+\tilde{y} \\ z_m=(r+\tilde{r})\sin(\theta+\tilde{\theta})=z+\tilde{z} \end{cases} \tag{2-48}$$

式中

$$x=r\cos(\varphi)\cos(\theta),\quad y=r\sin(\varphi)\cos(\theta),\quad z=r\sin(\theta)$$

令 $\boldsymbol{e}=[\tilde{x}\ \tilde{y}\ \tilde{z}]^{\mathrm{T}}$，则有

$$\boldsymbol{\mu}=E[\boldsymbol{e}|r,\varphi,\theta]=[\mu_x\ \mu_y\ \mu_z]^{\mathrm{T}}$$

$$\boldsymbol{R}=\operatorname{cov}[\boldsymbol{e}|r,\varphi,\theta]=\begin{bmatrix}\boldsymbol{R}_{xx} & \boldsymbol{R}_{xy} & \boldsymbol{R}_{xz}\\ \boldsymbol{R}_{yx} & \boldsymbol{R}_{yy} & \boldsymbol{R}_{yz}\\ \boldsymbol{R}_{zx} & \boldsymbol{R}_{zy} & \boldsymbol{R}_{zz}\end{bmatrix}$$

利用独立性条件及前文的结论，可导出如下结果：

$$\mu_x=r\cos\varphi\cos\theta(\mathrm{e}^{-\sigma_\varphi^2/2}\mathrm{e}^{-\sigma_\theta^2/2}-1)$$

$$\mu_y=r\sin\varphi\cos\theta(\mathrm{e}^{-\sigma_\varphi^2/2}\mathrm{e}^{-\sigma_\theta^2/2}-1)$$

$$\mu_z=r\sin\theta(\mathrm{e}^{-\sigma_\theta^2/2}-1)$$

$$\boldsymbol{R}_{xx}=\operatorname{var}[\tilde{x}|r,\varphi,\theta]=[r^2(a_x a_{xy}-\cos^2\varphi\cos^2\theta)+\sigma_r^2 a_x a_{xy}]\mathrm{e}^{-\sigma_\varphi^2}\mathrm{e}^{-\sigma_\theta^2}$$

$$\boldsymbol{R}_{yy}=\operatorname{var}[\tilde{y}|r,\varphi,\theta]=[r^2(a_y a_{xy}-\sin^2\varphi\cos^2\theta)+\sigma_r^2 a_y a_{xy}]\mathrm{e}^{-\sigma_\varphi^2}\mathrm{e}^{-\sigma_\theta^2}$$

$$\boldsymbol{R}_{zz}=\operatorname{var}[\tilde{z}|r,\varphi,\theta]=[r^2(a_z-\sin^2\theta)+\sigma_r^2 a_z]\mathrm{e}^{-\sigma_\theta^2}$$

$$\boldsymbol{R}_{yx}=\boldsymbol{R}_{xy}=\operatorname{cov}[\tilde{x},\tilde{y}|r,\varphi,\theta]=[r^2(a_{xy}-\mathrm{e}^{\sigma_\theta^2}\cos^2\theta)+\sigma_r^2 a_{xy}]\mathrm{e}^{-2\sigma_\varphi^2}\mathrm{e}^{-\sigma_\theta^2}\sin\varphi\cos\varphi$$

$$\boldsymbol{R}_{zx}=\boldsymbol{R}_{xz}=\operatorname{cov}[\tilde{x},\tilde{z}|r,\varphi,\theta]=[r^2(1-\mathrm{e}^{\sigma_\theta^2})+\sigma_r^2]\mathrm{e}^{-\sigma_\varphi^2/2}\mathrm{e}^{-2\sigma_\theta^2}\cos\varphi\sin\theta\cos\theta$$

$$\boldsymbol{R}_{zy}=\boldsymbol{R}_{yz}=\operatorname{cov}[\tilde{y},\tilde{z}|r,\varphi,\theta]=[r^2(1-\mathrm{e}^{\sigma_\theta^2})+\sigma_r^2]\mathrm{e}^{-\sigma_\varphi^2/2}\mathrm{e}^{-2\sigma_\theta^2}\sin\varphi\cos\theta\sin\theta$$

其中

$$a_x=\sin^2\varphi\sinh\sigma_\varphi^2+\cos^2\varphi\cosh\sigma_\varphi^2$$

$$a_y=\sin^2\varphi\cosh\sigma_\varphi^2+\cos^2\varphi\sinh\sigma_\varphi^2$$

$$a_z=\sin^2\theta\cosh\sigma_\theta^2+\cos^2\theta\sinh\sigma_\theta^2$$

$$a_{xy}=\sin^2\theta\sinh\sigma_\theta^2+\cos^2\theta\cosh\sigma_\theta^2$$

与二维情况类似，求解上述偏差要求目标真实距离、方位和俯仰信息已知。在这些信息未知的条件下，同样需要利用量测结果对误差均值和协方差求期望，即

$$\boldsymbol{\mu}_{\text{ave}}=E[\boldsymbol{\mu}|r_m,\varphi_m,\theta_m]=[\mu_{\text{ave},x}\ \mu_{\text{ave},y}\ \mu_{\text{ave},z}]^{\mathrm{T}} \tag{2-49}$$

$$\boldsymbol{R}_{\text{ave}}=E[\boldsymbol{R}|r_m,\varphi_m,\theta_m]=\begin{bmatrix}\boldsymbol{R}_{\text{ave},xx} & \boldsymbol{R}_{\text{ave},xy} & \boldsymbol{R}_{\text{ave},xz}\\ \boldsymbol{R}_{\text{ave},yx} & \boldsymbol{R}_{\text{ave},yy} & \boldsymbol{R}_{\text{ave},yz}\\ \boldsymbol{R}_{\text{ave},zx} & \boldsymbol{R}_{\text{ave},zy} & \boldsymbol{R}_{\text{ave},zz}\end{bmatrix} \tag{2-50}$$

利用式（2-47）对式（2-49）和式（2-50）中的各式进行展开，并利用三角关系式，得

$$\mu_{\text{ave},x}=r_m\cos\varphi_m\cos\theta_m(\mathrm{e}^{-\sigma_\varphi^2}\mathrm{e}^{-\sigma_\theta^2}-\mathrm{e}^{-\sigma_\varphi^2/2}\mathrm{e}^{-\sigma_\theta^2/2})$$

$$\mu_{\text{ave},y}=r_m\sin\varphi_m\cos\theta_m(\mathrm{e}^{-\sigma_\varphi^2}\mathrm{e}^{-\sigma_\theta^2}-\mathrm{e}^{-\sigma_\varphi^2/2}\mathrm{e}^{-\sigma_\theta^2/2})$$

$$\mu_{\text{ave},z}=r_m\sin\theta_m(\mathrm{e}^{-\sigma_\theta^2}-\mathrm{e}^{-\sigma_\theta^2/2})$$

$$\boldsymbol{R}_{\mathrm{ave},xx}=[r_m^2(b_x b_{xy}-a_x a_{xy})+\sigma_r^2(2b_x b_{xy}-a_x a_{xy})]\mathrm{e}^{-2\sigma_\varphi^2}\mathrm{e}^{-2\sigma_\theta^2}$$

$$\boldsymbol{R}_{\mathrm{ave},yy}=[r_m^2(b_y b_{xy}-a_y a_{xy})+\sigma_r^2(2b_y b_{xy}-a_y a_{xy})]\mathrm{e}^{-2\sigma_\varphi^2}\mathrm{e}^{-2\sigma_\theta^2}$$

$$\boldsymbol{R}_{\mathrm{ave},zz}=[r_m^2(b_z-a_z)+\sigma_r^2(2b_z-a_z)]\mathrm{e}^{-2\sigma_\theta^2}$$

$$\boldsymbol{R}_{\mathrm{ave},yx}=\boldsymbol{R}_{\mathrm{ave},xy}=[r_m^2(b_{xy}-a_{xy}\mathrm{e}^{\sigma_\varphi^2})+\sigma_r^2(2b_{xy}-a_{xy}e^{\sigma_\varphi^2})]\mathrm{e}^{-4\sigma_\varphi^2}\mathrm{e}^{-2\sigma_\theta^2}\sin\varphi_m\cos\varphi_m$$

$$\boldsymbol{R}_{\mathrm{ave},zx}=\boldsymbol{R}_{\mathrm{ave},xz}=[r_m^2(1-\mathrm{e}^{\sigma_\theta^2})+\sigma_r^2(2-\mathrm{e}^{\sigma_\theta^2})]\mathrm{e}^{-\sigma_\varphi^2}\mathrm{e}^{-4\sigma_\theta^2}\cos\varphi_m\sin\theta_m\cos\theta_m$$

$$\boldsymbol{R}_{\mathrm{ave},zy}=\boldsymbol{R}_{\mathrm{ave},yz}=[r_m^2(1-\mathrm{e}^{\sigma_\theta^2})+\sigma_r^2(2-\mathrm{e}^{\sigma_\theta^2})]\mathrm{e}^{-\sigma_\varphi^2}\mathrm{e}^{-4\sigma_\theta^2}\sin\varphi_m\sin\theta_m\cos\theta_m$$

其中

$$a_x=\sin^2\varphi_m\sinh\sigma_\varphi^2+\cos^2\varphi_m\cosh\sigma_\varphi^2$$

$$a_y=\sin^2\varphi_m\cosh\sigma_\varphi^2+\cos^2\varphi_m\sinh\sigma_\varphi^2$$

$$a_z=\sin^2\theta_m\cosh\sigma_\theta^2+\cos^2\theta_m\sinh\sigma_\theta^2$$

$$a_{xy}=\sin^2\theta_m\sinh\sigma_\theta^2+\cos^2\theta_m\cosh\sigma_\theta^2$$

$$b_x=\sin^2\varphi_m\sinh 2\sigma_\varphi^2+\cos^2\varphi_m\cosh 2\sigma_\varphi^2$$

$$b_y=\sin^2\varphi_m\cosh 2\sigma_\varphi^2+\cos^2\varphi_m\sinh 2\sigma_\varphi^2$$

$$b_z=\sin^2\theta_m\cosh 2\sigma_\theta^2+\cos^2\theta_m\sinh 2\sigma_\theta^2$$

$$b_{xy}=\sin^2\theta_m\sinh 2\sigma_\theta^2+\cos^2\theta_m\cosh 2\sigma_\theta^2$$

由此可得式（2-48）修正后的转换量测公式为

$$z_c=\begin{bmatrix}\boldsymbol{x}_{c,m}\\ \boldsymbol{y}_{c,m}\\ z_{c,m}\end{bmatrix}=\begin{bmatrix}r_m\cos\varphi_m\cos\theta_m\\ r_m\sin\varphi_m\cos\theta_m\\ r_m\sin\theta_m\end{bmatrix}-\begin{bmatrix}\mu_{ave,x}\\ \mu_{ave,y}\\ \mu_{ave,z}\end{bmatrix}$$

2.2 非线性滤波性能评估

2.2.1 通用误差评估指标

1. 均方根误差（RMSE）

假设第 m 次实验中状态量 $\boldsymbol{x}_k\in\mathbb{R}^{n_x}$ 的滤波估计为 $\hat{\boldsymbol{x}}_k^m$，则 M 次独立实验得到的 $\boldsymbol{x}_k$ 估计的均方根误差可表示为

$$RMSE_k[n]=\sqrt{\frac{1}{M}\sum_{m=1}^{M}(\boldsymbol{x}_k[n]-\hat{\boldsymbol{x}}_k^m[n])^2}\quad ,n=1,2,\cdots,n_x \tag{2-51}$$

式中：$\boldsymbol{x}_k[n]$ 表示状态矢量 $\boldsymbol{x}_k$ 的第 n 维。

2. 归一化估计方差（NEES）

设 k 时刻 $\boldsymbol{x}_k$ 的滤波估计误差为 $\tilde{\boldsymbol{x}}_k=\boldsymbol{x}_k-\hat{\boldsymbol{x}}_k$，其协方差矩阵为 $\boldsymbol{P}_k$，则归一化估计方差可表示为

$$\varepsilon_k=\tilde{\boldsymbol{x}}_k^{\mathrm{T}}(\boldsymbol{P}_k)^{-1}\tilde{\boldsymbol{x}}_k \tag{2-52}$$

在线性、高斯条件下，ε_k 服从自由度为 n_x 的卡方分布（$\chi^2_{n_x}$），此时 $E[\varepsilon_k]\approx\frac{1}{M}\sum_{m=1}^{M}\varepsilon_k^m=n_x$。该指标常用于评估滤波估计结果是否具有一致性。

3. 归一化平均估计误差（NMEE）

假设在 M 次实验中第 m 次的估计误差为 $\tilde{x}_k^m$，则归一化平均估计误差可表示为

$$\bar{u}_k[n]=\frac{1}{M}\sum_{m=1}^{M}\frac{\tilde{x}_k^m[n]}{\sqrt{P_k[n]}}\quad n=1,2,\cdots,n_x \tag{2-53}$$

在理想条件下，$\bar{u}_k[n]$ 服从零均值、方差为 $1/M$ 的高斯分布，该指标常用于测试状态矢量中单分量的估计一致性。

2.2.2 滤波误差理论下限

1. 目标状态估计误差下限

现实中用于非线性滤波的方法可能多种多样，尽管通过比较研究的手段可以评估各种方法的优劣，但若没有一个理论上的最优误差限，很难评价一种方法是否还有改进的余地或在多大程度上接近最优。由此可见，对理论误差限的分析具有重要的意义[24-26]。但需要指出的是，与传统参数估计的误差下限（称为克拉美—罗下限，CRLB）不同，由于系统为动态模型（含过程噪声），该条件下得到的状态估计误差限通常称为后验克拉美—罗限（PCRB）。下面将分析 PCRB 的计算问题，为此，重写式（1-1）和式（1-2）描述的非线性系统方程为

$$\boldsymbol{x}_k=f(\boldsymbol{x}_{k-1})+\boldsymbol{w}_k \tag{2-54}$$

$$\boldsymbol{y}_k=h(\boldsymbol{x}_k)+\boldsymbol{v}_k \tag{2-55}$$

其中，各变量所代表的含义与前文所述相同。

设 $\boldsymbol{x}_{0:k}=\{\boldsymbol{x}_i\}_{i=0}^{i=k}$ 和 $\boldsymbol{y}_{1:k}=\{\boldsymbol{y}_i\}_{i=1}^{i=k}$ 为到 k 时刻为止的累积状态序列和观测序列，则在任一时刻 k，$\boldsymbol{x}_{0:k}$ 和 $\boldsymbol{y}_{1:k}$ 的联合概率分布为

$$p_k=p(\boldsymbol{x}_{0:k},\boldsymbol{y}_{1:k})=p(\boldsymbol{x}_0)\prod_{i=1}^{k}p(\boldsymbol{y}_i\,|\,\boldsymbol{x}_i)\prod_{j=1}^{k}p(\boldsymbol{x}_j\,|\,\boldsymbol{x}_{j-1}) \tag{2-56}$$

记 $\boldsymbol{J}(\boldsymbol{x}_{0:k})$（维数为 $(n_xk)\times(n_xk)$）为上式对 $\boldsymbol{x}_{0:k}$ 的 Fisher 信息矩阵（FIM），$\boldsymbol{J}_k$（维数为 $n_x\times n_x$）为 $[\boldsymbol{J}(\boldsymbol{x}_{0:k})]^{-1}$ 的右下部分，则 $\boldsymbol{J}_k^{-1}$ 即为状态 $\boldsymbol{x}_k$ 的估计误差下限。

若将 $\boldsymbol{x}_{0:k}$ 分为两部分即 $\boldsymbol{x}_{0:k}=[\boldsymbol{x}_{0:k-1}^{\mathrm{T}}\ \boldsymbol{x}_k^{\mathrm{T}}]^{\mathrm{T}}$，则相应地

$$\boldsymbol{J}(\boldsymbol{x}_{0:k})=\begin{bmatrix}\boldsymbol{A}_k & \boldsymbol{B}_k\\ \boldsymbol{B}_k^T & \boldsymbol{C}_k\end{bmatrix}=\begin{bmatrix}E\{-\Delta_{\boldsymbol{x}_{0:k-1}}^{\boldsymbol{x}_{0:k-1}}\ln p_k\} & E\{-\Delta_{\boldsymbol{x}_{0:k-1}}^{\boldsymbol{x}_k}\ln p_k\}\\ E\{-\Delta_{\boldsymbol{x}_k}^{\boldsymbol{x}_{0:k-1}}\ln p_k\} & E\{-\Delta_{\boldsymbol{x}_k}^{\boldsymbol{x}_k}\ln p_k\}\end{bmatrix} \tag{2-57}$$

式中：∇ 和 Δ 分别为一阶和二阶偏导算子，即 $\nabla_\alpha\triangleq\left[\frac{\partial}{\partial\alpha_1}\frac{\partial}{\partial\alpha_2}\cdots\frac{\partial}{\partial\alpha_n}\right]^{\mathrm{T}}$，$\Delta_\beta^\alpha=\nabla_\beta\nabla_\alpha^{\mathrm{T}}$。

根据分块矩阵求逆定理，得

$$\boldsymbol{J}_k=\boldsymbol{C}_k-\boldsymbol{B}_k^{\mathrm{T}}\boldsymbol{A}_k^{-1}\boldsymbol{B}_k \tag{2-58}$$

由此可见，矩阵 $\boldsymbol{J}_k$ 的计算涉及 $\boldsymbol{A}_k$ 的求逆。很明显，随着 k 的不断增加，其运算过程极为复杂。为此，需要导出递推计算公式。注意到在 $k+1$ 时刻，有 $\boldsymbol{x}_{0:k+1}=[\boldsymbol{x}_{0:k-1}^{\mathrm{T}}\ \boldsymbol{x}_k^{\mathrm{T}}\ \boldsymbol{x}_{k+1}^{\mathrm{T}}]^{\mathrm{T}}$，

于是有

$$\begin{aligned}p_{k+1} &= p(\boldsymbol{x}_{0:k+1},\boldsymbol{y}_{0:k+1}) = p(\boldsymbol{x}_{0:k},\boldsymbol{y}_{0:k},\boldsymbol{x}_{k+1},\boldsymbol{y}_{k+1})\\ &= p(\boldsymbol{x}_{0:k},\boldsymbol{y}_{0:k})p(\boldsymbol{x}_{k+1}\mid \boldsymbol{x}_{0:k},\boldsymbol{y}_{0:k})p(\boldsymbol{y}_{k+1}\mid \boldsymbol{x}_{k+1},\boldsymbol{x}_{0:k},\boldsymbol{y}_{0:k})\\ &= p(\boldsymbol{x}_{0:k},\boldsymbol{y}_{0:k})p(\boldsymbol{x}_{k+1}\mid \boldsymbol{x}_{k})p(\boldsymbol{y}_{k+1}\mid \boldsymbol{x}_{k+1})\end{aligned} \tag{2-59}$$

$$\begin{aligned}\boldsymbol{J}(\boldsymbol{x}_{0:k+1}) &= \begin{bmatrix} E\{-\Delta_{\boldsymbol{x}_{0:k-1}}^{\boldsymbol{x}_{0:k-1}} \ln p_{k+1}\} & E\{-\Delta_{\boldsymbol{x}_{0:k-1}}^{\boldsymbol{x}_{k}} \ln p_{k+1}\} & E\{-\Delta_{\boldsymbol{x}_{0:k-1}}^{\boldsymbol{x}_{k+1}} \ln p_{k+1}\} \\ E\{-\Delta_{\boldsymbol{x}_{k}}^{\boldsymbol{x}_{0:k-1}} \ln p_{k+1}\} & E\{-\Delta_{\boldsymbol{x}_{k}}^{\boldsymbol{x}_{k}} \ln p_{k+1}\} & E\{-\Delta_{\boldsymbol{x}_{k}}^{\boldsymbol{x}_{k+1}} \ln p_{k+1}\} \\ E\{-\Delta_{\boldsymbol{x}_{k+1}}^{\boldsymbol{x}_{0:k-1}} \ln p_{k+1}\} & E\{-\Delta_{\boldsymbol{x}_{k+1}}^{\boldsymbol{x}_{k}} \ln p_{k+1}\} & E\{-\Delta_{\boldsymbol{x}_{k+1}}^{\boldsymbol{x}_{k+1}} \ln p_{k+1}\} \end{bmatrix}\\ &= \left[\begin{array}{cc|c} \boldsymbol{A}_k & \boldsymbol{B}_k & 0 \\ \boldsymbol{B}_k^{\mathrm{T}} & \boldsymbol{C}_k + \boldsymbol{D}_k^{11} & \boldsymbol{D}_k^{12} \\ \hline 0 & \boldsymbol{D}_k^{21} & \boldsymbol{D}_k^{22} \end{array}\right]\end{aligned} \tag{2-60}$$

式中

$$\boldsymbol{D}_k^{11} = E\{-\Delta_{\boldsymbol{x}_k}^{\boldsymbol{x}_k} \ln p(\boldsymbol{x}_{k+1}\mid \boldsymbol{x}_k)\} \tag{2-61}$$

$$\boldsymbol{D}_k^{12} = E\{-\Delta_{\boldsymbol{x}_k}^{\boldsymbol{x}_{k+1}} \ln p(\boldsymbol{x}_{k+1}\mid \boldsymbol{x}_k)\} \tag{2-62}$$

$$\boldsymbol{D}_k^{21} = [\boldsymbol{D}_k^{12}]^{\mathrm{T}} = E\{-\Delta_{\boldsymbol{x}_{k+1}}^{\boldsymbol{x}_k} \ln p(\boldsymbol{x}_{k+1}\mid \boldsymbol{x}_k)\} \tag{2-63}$$

$$\boldsymbol{D}_k^{22} = E\{-\Delta_{\boldsymbol{x}_{k+1}}^{\boldsymbol{x}_{k+1}} \ln p(\boldsymbol{x}_{k+1}\mid \boldsymbol{x}_k)\} + E\{-\Delta_{\boldsymbol{x}_{k+1}}^{\boldsymbol{x}_{k+1}} \ln p(\boldsymbol{y}_{k+1}\mid \boldsymbol{x}_{k+1})\} \tag{2-64}$$

同理，FIM 子阵 $\boldsymbol{J}_{k+1}$ 对应于 $[\boldsymbol{J}(\boldsymbol{x}_{0:k+1})]^{-1}$ 右下角位置维数为 $n_x \times n_x$ 的块矩阵，对照式（2-58），此时有

$$\boldsymbol{J}_{k+1} = \boldsymbol{D}_k^{22} - \begin{bmatrix} \boldsymbol{0} & \boldsymbol{D}_k^{21} \end{bmatrix} \begin{bmatrix} \boldsymbol{A}_k & \boldsymbol{B}_k \\ \boldsymbol{B}_k^{T} & \boldsymbol{C}_k + \boldsymbol{D}_k^{11} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{D}_k^{12} \end{bmatrix} = \boldsymbol{D}_k^{22} - \boldsymbol{D}_k^{21}(\boldsymbol{J}_k + \boldsymbol{D}_k^{11})^{-1}\boldsymbol{D}_k^{12} \tag{2-65}$$

式（2-65）即为一般非线性系统滤波问题所对应的信息矩阵的递推关系式，由此可求得状态估计的后验 PCRB 满足 $\boldsymbol{P}_k \geqslant \boldsymbol{J}_k^{-1}$。

下面具体分析 PCRB 的计算问题，主要分两种重要情况：

1）高斯噪声情况

在高斯噪声假设条件下，可设 $\boldsymbol{w}_k=\mathcal{N}(\boldsymbol{w}_k;\boldsymbol{0},\boldsymbol{Q}_k)$、$\boldsymbol{v}_k=\mathcal{N}(\boldsymbol{v}_k;\boldsymbol{0},\boldsymbol{R}_k)$，初始状态分布为 $\boldsymbol{x}_0=\mathcal{N}(\boldsymbol{x}_0;\hat{\boldsymbol{x}}_0,P_0)$，此时有

$$p(\boldsymbol{x}_{k+1}\mid \boldsymbol{x}_k)=\mathcal{N}(\boldsymbol{x}_{k+1};f(\boldsymbol{x}_k),\boldsymbol{Q}_k)$$

$$p(z_{k+1}\mid \boldsymbol{x}_{k+1})=\mathcal{N}(z_{k+1};h(\boldsymbol{x}_{k+1}),\boldsymbol{R}_{k+1})$$

于是有

$$\boldsymbol{D}_k^{11} = E[\nabla_{\boldsymbol{x}_k} f^{\mathrm{T}}(\boldsymbol{x}_k)\boldsymbol{Q}_k^{-1}\nabla_{\boldsymbol{x}_k} f(\boldsymbol{x}_k)]$$

$$\boldsymbol{D}_k^{12} = -E[\nabla_{\boldsymbol{x}_k} f^{\mathrm{T}}(\boldsymbol{x}_k)]\boldsymbol{Q}_k^{-1} = [\boldsymbol{D}_k^{21}]^{\mathrm{T}}$$

$$\boldsymbol{D}_k^{22} = \boldsymbol{Q}_k^{-1} + E\{\nabla_{\boldsymbol{x}_{k+1}} h^{\mathrm{T}}(\boldsymbol{x}_{k+1})\boldsymbol{R}_{k+1}^{-1}h(\boldsymbol{x}_{k+1})\}$$

上述几式中都存在求期望运算，实际应用时常常用蒙特卡罗方法进行近似。假设进行 M 次蒙特卡罗实验，$\{\boldsymbol{x}_k^m\}_{k=0}^N$ 表示第 $m(m=1,2,\cdots,M)$ 次实验中状态轨迹的样本，则

$$\boldsymbol{D}_k^{11} = \frac{1}{M}\sum_{m=1}^{M}\nabla_{\boldsymbol{x}_k} f^{\mathrm{T}}(\boldsymbol{x}_k)\boldsymbol{Q}_k^{-1}\nabla_{\boldsymbol{x}_k} f^{\mathrm{T}}(\boldsymbol{x}_k)\Big|_{\boldsymbol{x}_k=\boldsymbol{x}_k^m} \tag{2-66}$$

$$D_k^{12} = -\frac{1}{M}\sum_{m=1}^{M}\nabla_{x_k} f^{\mathrm{T}}(x_k)Q_k^{-1}\Big|_{x_k = x_k^m} = [D_k^{21}]^{\mathrm{T}} \tag{2-67}$$

$$D_k^{22} = Q_k^{-1} + \frac{1}{M}\sum_{m=1}^{M}\nabla_{x_{k+1}} h^{\mathrm{T}}(x_{k+1})R_{k+1}^{-1}\nabla_{x_{k+1}} h(x_{k+1})\Big|_{x_{k+1} = x_{k+1}^m} \tag{2-68}$$

2）线性、高斯条件

进一步假设式（2-54）和式（2-55）所描述的是线性系统，即 $f(x_{k-1})=F_k x_{k-1}$，$h(x_k)=H_k x_k$，此时根据式（2-61）～式（2-65），得

$$J_{k+1} = H_{k+1}^{\mathrm{T}} R_{k+1}^{-1} H_{k+1} + Q_k^{-1} - Q_k^{-1} F_k (J_k + F_k^{\mathrm{T}} Q_k^{-1} F_k)^{-1} F_k^{\mathrm{T}} Q_k^{-1}$$

令 $P_k = J_k^{-1}$ 并利用矩阵引理，得

$$\begin{aligned} P_{k+1}^{-1} &= (Q_k + F_k P_k F_k^{\mathrm{T}})^{-1} + H_{k+1}^{\mathrm{T}} R_{k+1}^{-1} H_{k+1} \\ &= P_{k+1|k}^{-1} + H_{k+1}^{\mathrm{T}} R_{k+1}^{-1} H_{k+1} \end{aligned}$$

由此可见，该式正是卡尔曼滤波算法中的协方差矩阵表达式（以逆矩阵的形式给出，对应文献[10]中的信息滤波（Information Filtering）公式。这也正好说明线性、高斯条件下的卡尔曼滤波是最优估计，其估计误差协方差矩阵正好对应理想条件下的理论误差下限。

2. 混合状态估计误差下限

考虑更一般情况下的误差限计算问题，即状态中包含一未知的常数参量 θ，这一问题在实际应用中具有普遍性（如匀转弯目标的转弯率、匀加速目标的加速度以及扩展目标的尺度大小等）。此时，系统方程可记为

$$x_k = f(x_{k-1}, \theta) + w_k \tag{2-69}$$

$$y_k = h(x_k, \theta) + v_k \tag{2-70}$$

采用与式（2-56）类似的方法，在任一时刻 k，$x_{0:k}$、$y_{1:k}$ 与 θ 的联合概率分布为

$$\begin{aligned} p_k = p(x_{0:k}, y_{1:k}, \theta) &= \underset{y_{1:k}|x_{0:k}}{p(y_{1:k}|x_{0:k})}\ \underset{x_{0:k}|\theta}{p(y_{1:k}|x_{0:k})}\ p(\theta) \\ &= p(x_0)p(\theta)\prod_{i=1}^{k} p(y_i \mid x_i, \theta)\prod_{j=1}^{k} p(x_j \mid x_{j-1}, \theta) \end{aligned} \tag{2-71}$$

对式（2-71）取对数，得

$$\begin{aligned} \ln p(x_{0:k}, y_{1:k}, \theta) &= \ln \underset{y_{1:k}|x_{0:k}}{p(y_{1:k}|x_{0:k})}\ \underset{x_{0:k}|\theta}{p(y_{1:k}|x_{0:k})}\ p(\theta) \\ &= \ln p(x_0) + \ln p(\theta) + \sum_{i=0}^{k} p(y_i \mid x_i, \theta) + \sum_{i=1}^{k} p(x_j \mid x_{j-1}, \theta) \end{aligned} \tag{2-72}$$

定义

$$\begin{aligned} K_{i+1}^{\theta i} &= E\{-\nabla_{x_i}[\nabla_{\theta} \ln p(x_{i+1} \mid x_i, \theta)]^{\mathrm{T}}\} \\ K_{i+1}^{\theta, i+1} &= E\{-\nabla_{x_{i+1}}[\nabla_{\theta} \ln p(x_{i+1} \mid x_i, \theta)]^{\mathrm{T}}\} \\ K_{i+1}^{\theta} &= E\{-\nabla_{\theta}[\nabla_{\theta} \ln p(x_{i+1} \mid x_i, \theta)]^{\mathrm{T}}\} \\ L_i^{\theta i} &= E\{-\nabla_{x_i}[\nabla_{\theta} \ln p(z_i \mid x_i, \theta)]^{\mathrm{T}}\} \end{aligned}$$

$$L_i^{\theta} = E\{-\nabla_{\theta}[\nabla_{\theta} \ln p(z_i \mid x_i, \theta)]^{\mathrm{T}}\}$$

$$A_0 = E\{-\nabla_{\theta}[\nabla_{\theta} \ln p(\theta)]^{\mathrm{T}}\}$$

$$A_k = A_0 + L_0^{\theta} + \sum_{i=1}^{k-1}(L_i^{\theta} + L_i^{\theta}) + K_k^{\theta}$$

$$G_i = L_i^{\theta i} + K_i^{\theta i} + K_{i+1}^{\theta i}$$

式中：$i=0,1,\cdots,k$，$L_i^{\theta i}=[L_i^{\theta i}]^{\mathrm{T}}$，$K_i^{\theta i}=[K_i^{\theta i}]^{\mathrm{T}}$，$K_{i+1}^{\theta i}=[K_{i+1}^{\theta i}]^{\mathrm{T}}$；矩阵 $K_i^{\theta i}$、$K_{i+1}^{\theta i}$、$L_i^{\theta i}$ 维数大小为 $m\times n$；矩阵 K_{i+1}^{θ}、L_i^{θ}、A_k 的维数为 $m\times m$，当 i=0 时对应的 $K_0^{\theta 0}$ 为 $m\times n$ 阶零矩阵。

此时，对应联合状态量 $(x_{0:k},\theta)$ 的 FIM 为

$$J_k = -E\{-\nabla_{x_{0:k},\theta}[\nabla_{x_{0:k},\theta} \ln p(x_{0:k}, z_{1:k}, \theta)]^{\mathrm{T}}\}$$

$$= \begin{bmatrix} J_k^{11} & J_k^{12} & [G^{k-1}]^{\mathrm{T}} \\ J_k^{21} & J_k^{22} & L_k^{k\theta} + K_k^{k\theta} \\ G^{k-1} & L_k^{\theta k} + K_k^{\theta k} & L_k^{\theta} + A_k \end{bmatrix} \tag{2-73}$$

式中

$$G^i = [G_0 \ G_1 \cdots G_i]$$

于是，滤波估计的误差下限 C_k 为 J_k^{-1} 右下部分大小为 $(n+m)\times(n+m)$ 所对应的块矩阵（具体推导见文献[25]），且 C_k^{-1} 中的块元素满足下列递推关系式

$$J_k^{xx} = J_{k|k-1}^{xx} + L_k^k \tag{2-74}$$

$$J_k^{x\theta} = J_{k|k-1}^{x\theta} + L_k^{k\theta} \tag{2-75}$$

$$J_k^{\theta\theta} = J_{k|k-1}^{\theta\theta} + L_k^{\theta} \tag{2-76}$$

其中，一步预测 FIM 可表示为

$$J_{k+1|k}^{xx} = K_{k+1}^{k+1} - K_{k+1}^{k+1,k} \Delta_{kk}^{-1} K_{k+1}^{k,k+1} \tag{2-77}$$

$$J_{k+1|k}^{x\theta} = K_{k+1}^{k+1,\theta} - K_{k+1}^{k+1,k} \Delta_{kk}^{-1} \Delta_{kk}^{x\theta} \tag{2-78}$$

$$J_{k+1|k}^{\theta\theta} = J_k^{\theta\theta} + K_{k+1}^{\theta} - \Delta_{kk}^{\theta x} \Delta_{kk}^{-1} \Delta_{kk}^{x\theta} \tag{2-79}$$

递推算式（2-74）～式（2-79）的初始条件为 $J_{0|-1}^{xx}=K_0^0$，$J_{0|-1}^{x\theta}=K_0^{0\theta}=\mathbf{0}$，$J_{0|-1}^{\theta\theta}=A_0$。

值得一提的是，根据系统条件的不同，可能存在不同的 PCRB 计算方法[27,28]；另外，在相关噪声条件下，在求解 PCRB 过程中可能面临矩阵无法直接计算问题，此时需要进行一些特殊的处理，关于这一点可参阅文献[24]。

2.3 应用实例

2.3.1 二维转换量测目标跟踪

假设二维平面中的目标运动方程满足

$$x_k = Fx_{k-1} + Gw_{k-1} \tag{2-80}$$

式中：$x_k=[x_k\ \dot{x}_k\ y_k\ \dot{y}_k]^{\mathrm{T}}$ 为 k 时刻目标的状态；F 状态转移矩阵；G 为噪声输入矩阵；w_{k-1}

为零均值高斯过程噪声，且

$$\boldsymbol{F}=\begin{bmatrix}1 & T & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 1 & T\\0 & 0 & 0 & 1\end{bmatrix},\quad \boldsymbol{G}=\begin{bmatrix}T^2/2 & 0\\T & 0\\0 & T^2/2\\0 & T\end{bmatrix},\quad \operatorname{cov}[\boldsymbol{w}_k]=\boldsymbol{Q}_k=\begin{bmatrix}q_x^2 & 0\\0 & q_y^2\end{bmatrix}$$

设观测量包含目标距离和方位，观测方程为

$$\boldsymbol{z}_k=\begin{bmatrix}r_k\\\varphi_k\end{bmatrix}=\boldsymbol{h}(\boldsymbol{x}_k)+\boldsymbol{v}_k=\begin{bmatrix}\sqrt{x_k^2+y_k^2}\\\arctan(y_k/x_k)\end{bmatrix}+\begin{bmatrix}v_{r,k}\\v_{\varphi,k}\end{bmatrix} \tag{2-81}$$

式中：$v_{r,k}$、$v_{\varphi,k}$ 为零均值高斯测量噪声，$\operatorname{var}[v_{r,k}]=\sigma_r^2$，$\operatorname{var}[v_{\varphi,k}]=\sigma_\varphi^2$。

采用二维去偏转换方法进行处理，得

$$\boldsymbol{z}_{k,c}=\begin{bmatrix}x_{k,c}\\y_{k,c}\end{bmatrix}=\begin{bmatrix}r_k\cos\varphi_k\\r_k\sin\varphi_k\end{bmatrix}-\begin{bmatrix}\mu_{k,x}\\\mu_{k,y}\end{bmatrix} \tag{2-82}$$

式中

$$\mu_{k,x}=r_k\cos\varphi_k(\mathrm{e}^{-\sigma_\varphi^2}-\mathrm{e}^{-\sigma_\varphi^2/2}),\quad \mu_{k,y}=r_k\sin\varphi_k(\mathrm{e}^{-\sigma_\varphi^2}-\mathrm{e}^{-\sigma_\varphi^2/2})$$

于是，去偏转换测量模型可表示为

$$\boldsymbol{z}_k=\boldsymbol{H}\boldsymbol{x}_k+\boldsymbol{v}_{c,k} \tag{2-83}$$

式中：$\boldsymbol{H}$ 为观测矩阵；$\boldsymbol{v}_{c,k}$ 为去偏转换量测噪声，且

$$\boldsymbol{H}=\begin{bmatrix}1 & 0 & 0 & 0\\0 & 0 & 1 & 0\end{bmatrix},\quad \operatorname{cov}[\boldsymbol{v}_{c,k}]=\boldsymbol{R}_{c,k} \tag{2-84}$$

$\boldsymbol{R}_{c,k}$ 的计算方法与式（2-46）相同。

假设 T=2s，$q_x=q_y$=2m/s^2，目标初始位置和速度分别为 (60,80)km 和 $(-100, 100\sqrt{3})$m/s，初始位置和速度估计偏差分别为 (30,40)m、$(-5,4)$m/s，初始估计误差协方差矩阵

$$\boldsymbol{P}_0=\frac{1}{N}\sum_{n=1}^{N}\zeta_n\zeta_n^{\mathrm{T}}$$

式中

$$N=10000,\quad \zeta_n\sim\mathcal{N}(\boldsymbol{x};\boldsymbol{0},\boldsymbol{\Sigma}),\quad \boldsymbol{\Sigma}=\operatorname{diag}([200^2\ 50^2\ 200\ 50^2])$$

进一步假定 σ_r=20m，$\sigma_\varphi=1^\circ$，采用去偏转换卡尔曼滤波（CMKF）对目标进行跟踪，所得结果如图 2.6 所示。

由图 2.6 中的结果可以看出，经去偏转换处理后，直接采用线性条件下的卡尔曼滤波算法即可得到良好的跟踪效果；且在相同的距离、方位观测误差条件下，跟踪误差并没有随目标距离的变化而发生明显的变化。由图中的结果同样可见，与由直接转换量测得到的结果（图中 o 型线）相比，经滤波处理后目标位置估计误差明显减小，定位精度显著提高。

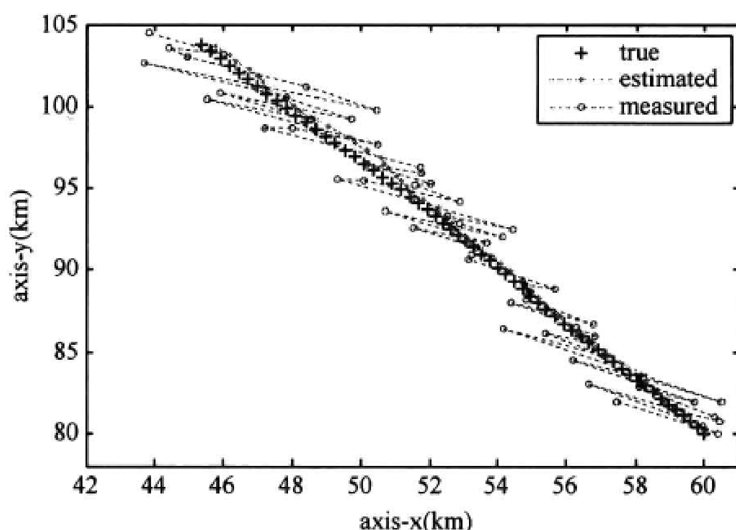


图 2.6 CMKF 目标跟踪示例

2.3.2 跟踪误差理论下限分析

仍假设二维平面中的目标运动方程满足

$$x_k = Fx_{k-1} + w_{k-1} \tag{2-85}$$

式中：F 状态转移矩阵；w_{k-1} 为零均值、方差为 Q_k 的高斯过程噪声，且

$$F=\begin{bmatrix} F_1 & 0 \\ 0 & F_1 \end{bmatrix},\quad Q_k=q^2\cdot\begin{bmatrix} Q_1 & 0 \\ 0 & Q_1 \end{bmatrix}$$

$$F_1=\begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix},\quad Q_1=\begin{bmatrix} T^3/3 & T^2/2 \\ T^2/2 & T \end{bmatrix}$$

设观测量包含目标距离和方位，则观测方程为

$$z_k=\begin{bmatrix} r_k \\ \varphi_k \end{bmatrix}=h_{2D}(x_k)+v_k=\begin{bmatrix} \sqrt{x_k^2+y_k^2} \\ \arctan(y_k/x_k) \end{bmatrix}+\begin{bmatrix} v_{r,k} \\ v_{\varphi,k} \end{bmatrix} \tag{2-86}$$

若增加径向距离观测，可得观测方程

$$z_k=\begin{bmatrix} r_k \\ \varphi_k \\ \dot{r}_k \end{bmatrix}=h_{3D}(x_k)+v_k=\begin{bmatrix} \sqrt{x_k^2+y_k^2} \\ \arctan(y_k/x_k) \\ \dfrac{x_k\dot{x}_k+y_k\dot{y}_k}{\sqrt{x_k^2+y_k^2}} \end{bmatrix}+\begin{bmatrix} v_{r,k} \\ v_{\varphi,k} \\ v_{\dot{r},k} \end{bmatrix} \tag{2-87}$$

式中：$v_{r,k}$、$v_{\varphi,k}$ 和 $v_{\dot{r},k}$ 分别为零均值高斯测量噪声，$\mathrm{var}[v_{r,k}]=\sigma_r^2$，$\mathrm{var}[v_{\varphi,k}]=\sigma_\varphi^2$，$\mathrm{var}[v_{\dot{r},k}]=\sigma_{\dot{r}}^2$。

假设 T=0.25s，q=2m/s^2，目标初始位置和速度分别为 (20,20)km 和 (60, − 200)m/s，初始估计（先验）误差协方差矩阵 P_0=diag([300^2 150^2 300 150^2])，σ_r=50m，σ_φ=1°，$\sigma_{\dot{r}}$=0.5m/s，在两种不同观测模型条件下利用式（2-66）～式（2-68）分别计算估计误差的理论下限，此时有

$$\frac{\partial h_{2D}(\boldsymbol{x})}{\partial \boldsymbol{x}}=\begin{bmatrix} -\frac{y}{r^2} & 0 & \frac{x}{r^2} & 0 \\ \frac{x}{r} & 0 & \frac{y}{r} & 0 \end{bmatrix},\quad r=\sqrt{x^2+y^2} \tag{2-88}$$

$$\frac{\partial h_{3D}(\boldsymbol{x})}{\partial \boldsymbol{x}}=\begin{bmatrix} -\frac{y}{r^2} & 0 & \frac{x}{r^2} & 0 \\ \frac{x}{r} & 0 & \frac{y}{r} & 0 \\ \frac{\dot{x}r-ax/r}{r^2} & \frac{x}{r} & \frac{\dot{y}r-ay/r}{r^2} & \frac{y}{r} \end{bmatrix},\quad a=x\dot{x}+y\dot{y} \tag{2-89}$$

图 2.7～图 2.10 给出了 500 次蒙特卡罗实验得到的两种不同观测条件下的状态估计误差理论下限，即 $\left[\sqrt{\boldsymbol{J}_k^{-1}}\right]_{i,i}$ $(i=1,2,3,4)$。由图中的结果可以看出，与传统的距离—方位跟踪相比，引入距离变化率观测信息可改善估计精度，同时提高误差收敛速度。这相当于在观测目标时增加了新的约束条件，从而增加了目标状态估计的确定性。

不过，若提高距离和方位的观测精度（如 σ_r=20m，σ_φ=0.5°），由图 2.11 和图 2.12 中的结果可知，由距离变化率信息带来的位置估计性能改善会被弱化，但其对目标速度估计精度的提高仍然是明显的（尤其在初始跟踪阶段），这与式（2-89）中包含了目标速度信息的事实是一致的。

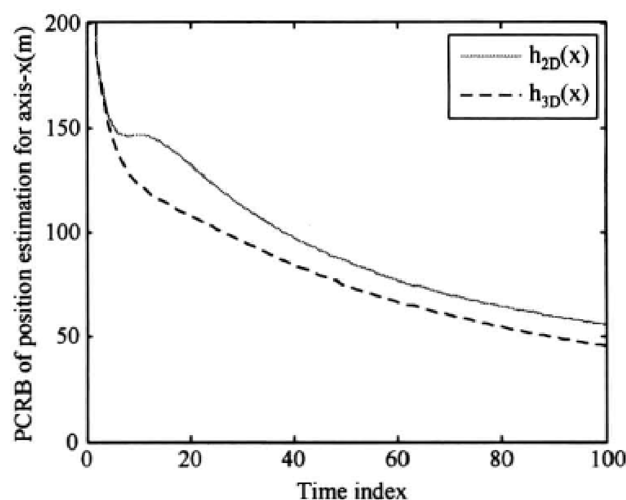


图 2.7　x 轴方向位置估计误差

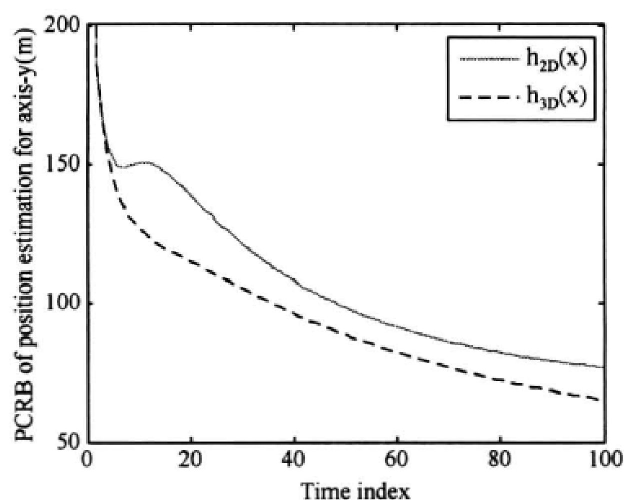


图 2.8　y 轴方向位置估计误差

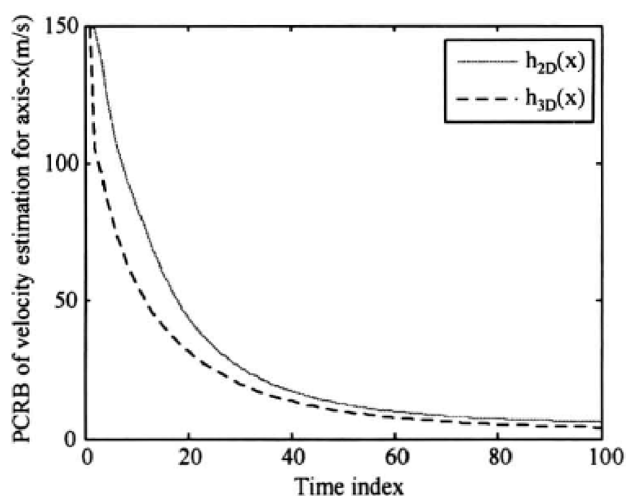


图 2.9　x 轴方向速度估计误差

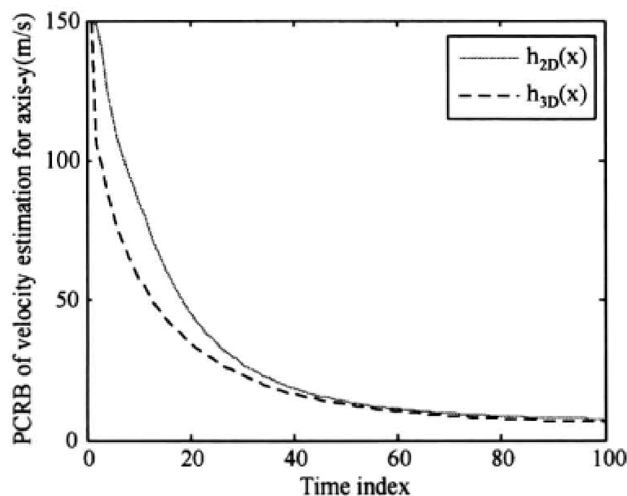


图 2.10　y 轴方向速度估计误差

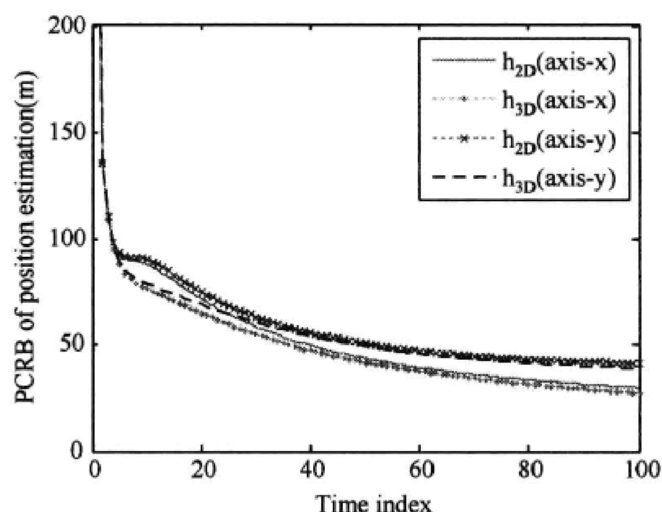


图 2.11　位置估计误差下限

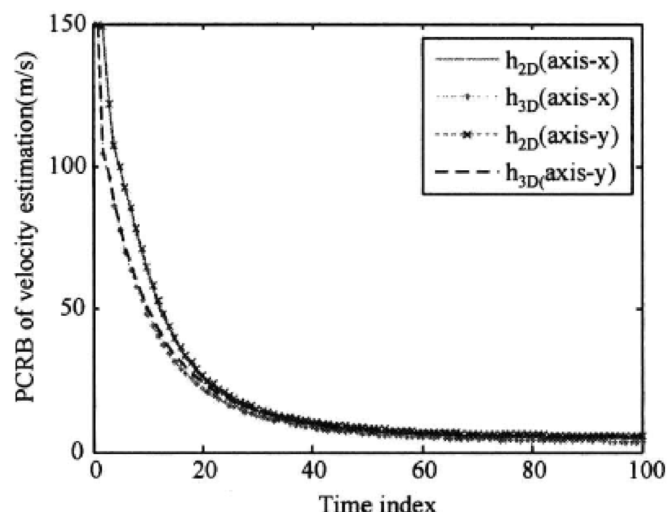


图 2.12　速度估计误差下限

2.4　本 章 小 结

本章主要总结了非线性跟踪中常用的目标运动模型及传感器观测模型，并阐述了非线性滤波的一些通用性能评价指标。

作为一类重要的传感器，雷达通常能获取目标方位和距离观测信息，但这些观测是非线性的，且在相同的观测精度下，目标离雷达越远，雷达对目标的实际定位精度就越差。解决这一问题的一条有效途径是采用去偏转换方法，将非线性量测转换为线性量测模型，进而采用卡尔曼滤波算法进行跟踪，本章对去偏转换方法进行了分析，并给出了应用实例。

克拉美—罗限表征了状态估计问题所能达到的误差理论下限，对评估各种跟踪算法的性能具有参考性、示范性的作用。针对随机系统中的后验克拉美—罗限（PCRB）分析问题，本章阐述了非线性条件下的误差下限计算方法，并指出线性、高斯系统特殊条件下信息滤波算法与 PCRB 的关系，在此基础上还分析了混合系统状态（系统状态中含未知参量）的 PCRB 计算问题，最后结合目标跟踪实例，比较了不同量测集下的跟踪误差下限。

参 考 文 献

[1] Jazwinski A H. Stochastic Processes and Filtering Theory[M]. New York：Academic Press，1970.

[2] Gelb A. Applied Optimal Estimation[M]. MA：MIT Press，1974.

[3] Anderson B D O，Moore J B. Optimal Filtering[M]. New Jersey: Prentice-Hall，1979.

[4] Maybeck P S. Stochastic Models，Estimation and Control[M]. New York：Academic Press，1982.

[5] Lewis F L. Optimal Estimation[M]. New York:John Wiley & Sons，1986.

[6] Kay S M. Fundamentals of Statistical Signal Processing：Estimation Theory[M]. New Jersey：Prentice-Hall，1998.

[7] Buey R S. Bayes theorem and digital realization for nonlinear filters[J]. Journal of Astronautic Sciences，1969，17(2)：80–94.

[8] Bar-shalom Y，X R Li，Kirubarajan T. Estimation with Application to Tracking and Navigation：Theory，Algorithm，and Software[M]. New York：John Wiley & Sons，2001.

[9] Ristic B，Arulampalam S，Gordon N，Beyond the Kalman：Particle filters for Tracking Applications[M]. London：Artech House，2004.

[10] Simon D. Optimal State Estimation Kalman，H_∞，and Nonlinear Apporaches[M]. New Jersey：John Wiley & Sons，2006.

[11] Farina A，Benvenuti D，Ristic B. A comparative study of the Benes filtering problem[J]. Signal Processing，2002，82：133–147.

[12] 周宏仁，敬忠良，王培德. 机动目标跟踪[M]. 北京：国防工业出版社，1991.

[13] X R Li，Jilkov V P. Survey of maneuvering target tracking. Part I：Dynamic models[J]. IEEE Transactions on Aerospace and Electronic Systems，2003，39(4)：1333–1364.

[14] Salmond D J，Birch H. A particle filter for track-before-detect[C]. Proceedings of the American Control Conference，2001：3755–3060.

[15] Boers Y，Driessen J N. Multitarget particle filter track before detect application[J]. IEE Proceedings-Radar Sonar Navigation，2004，151(6)：351–357.

[16] Boers Y，Driessen H，Torstensson J，et al. Track-before-detect algorithm for tracking extended targets[J]. IEE Proceedings-Radar Sonar Navigation，2006，153(4)：345–351.

[17] X Z Wang，Scala B L，Ellem R. Feature aided probabilistic data association for multi-target tracking [C]. Proceedings of 11th International Conference on Information Fusion，2008：1–7.

[18] 占荣辉，张军. 基于距离像长度特征辅助的雷达目标跟踪[J]，信号处理，2010，26(2)：166–169.

[19] 占荣辉，王威，张军. 利用方向角信息辅助雷达目标跟踪[J]. 系统工程与电子技术，2012，34(7)：1350–1353.

[20] Salmond D J，Parr M C. Track maintenance using measurements of target extent[J]. IEE Proceedings-Radar Sonar Navigation，2003，150(6)：389–395.

[21] Lifeng X，X. R Li. Hybrid Cramer–Rao lower bound on tracking ground moving extended target[C]. Proceedings of the International Conference on Information Fusion，2009，1037–1044.

[22] Lerro D，Bar-Shalom Y. Tracking with debiased consistent converted measurements versus EKF[J]. IEEE Transactions on Aerospace and Electronic Systems，1993，29(3)：1015–1022.

[23] Piotr S. Explicit expressions for debiased statistics of 3D converted measurements[J]. IEEE Transactions on Aerospace and Electronic Systems，1999，35(1)：368–370.

[24] Tichavsky P，Muravchik C H，Nehorai A. Posterior Cramér-Rao bounds for discrete-time nonlinear filtering[J]，IEEE Transactions on Signal Processing，1998，46(5)：1386–1396.

[25] Simandl M，Kralovec J，Tichavsky P. Filtering，predictive，and smoothing Cramér–Rao bounds for discrete-time nonlinear dynamic systems[J]，Automatica，2001，37：1703– 1716.

[26] Jun Z，Ronghui Z. PCRB analysis for passive target tracking[J]. Journal of Electronics，2008，25(1)：84–88.

[27] Zhiwen Z，Huadong M，Xiqin W. A comparison of posterior Cramer–Rao bounds for point and extended target tracking[J]. IEEE Signal Processing Letters，2010，17(10)：819–822.

[28] Esra S，Aydin A. Posterior Cramer–Rao lower bounds for dual Kalman estimation[J]. Digital Signal Processing，2012，22：47–53.

第二部分　解析高斯近似滤波与应用

贝叶斯推演为动态系统的状态估计问题提供了统一的解决方案，但是由于其最优估计的求解过程需要传播整个概率密度，这只有在线性、高斯条件下才能解析递推。对于一般的非线性/非高斯系统，通常难以得到解析递推表达式，求解过程必须采用某种近似手段。若将系统状态的概率密度等效近似为服从高斯或混合高斯分布，则由此得到的滤波器常被称为高斯滤波器或混合高斯滤波器（或高斯和滤波器）。

目前存在多种高斯滤波器，其主要区别在于近似手段的不同，但都是在卡尔曼滤波的框架下得到的（即隐含了对高斯噪声的假设）。常见的近似手段可归纳为三大类：①函数近似法，其主要思想是用非线性系统函数的低阶展开来近似非线性函数本身，典型的有扩展卡尔曼滤波器（EKF）、中心差分卡尔曼滤波器（CDKF）；②确定性采样近似法，用确定性样点来近似系统状态及其非线性系统传递后的分布，典型的有不敏卡尔曼滤波器（UKF）；③求积近似法，用各种数值手段来近似贝叶斯滤波递推式中的多维积分，典型的有高斯—厄米特卡尔曼滤波器（GHKF）、求容积卡尔曼滤波器（CKF）等。

上述提到的几种近似手段，都无一例外地将系统的状态分布近似单个多维高斯随机变量（GRV），这种近似对于常见的单模态（Modality）问题是合理的，但对于多模态（Multimodal）问题（如含有多个不同的系统状态模型/模式，具有非高斯过程/量测噪声等）这种近似方法显得无能为力，这就是为什么上述几种近似方法通常只用于求解非线性、高斯系统中的滤波问题的原因。解决多模态问题的一种有效途径是采用多个不同的高斯分量进行混合近似，这种方法的理论基础是任何一种复杂的分布都可能用混合高斯分量的形式达到任意精度的逼近。在这种理论指导下，各种混合高斯滤波（GMF）的方法也就应运而生，典型的有 GM-EKF、GM-UKF 等。

本部分主要阐述基于解析高斯/混合高斯近似的非线性滤波方法及其在目标跟踪中的应用问题。第 3 章根据函数近似原理，重点介绍了两种重要的近似方法，即泰勒展开近似和函数插值近似，并分析了基于这两种近似方法的滤波算法。第 4 章从函数状态分布的近似入手，介绍了不敏变换的原理，导出了卡尔曼框架下的滤波算法及其算法的衍生形式。第 5 章则从多维非线性函数的积分式求解这一角度，分析了求积型卡尔曼滤波的算法和实现方法。第 6 章针对多模态问题，系统介绍了基于混合高斯近似的滤波原理及通过 EKF、UKF 等来的实现的方法。每一章的最后均给出了具体的应用实例。

第 3 章　函数近似高斯滤波

通过第 1 章的介绍可知，卡尔曼滤波（KF）为线性、高斯条件下的递推贝叶斯估计问题提供了最优解的形式。显然，在一般的非线性系统中，满足最优解的条件不再成立。大量的研究将致力于寻求各种不同的近似途径得到次优解，其中一种最直接、自然的途径就是对非线性系统函数进行近似化处理。

对非线性函数的近似化处理通常有两种重要的方法：一种是利用泰勒级数展开（Taylor Series Expansion，TSE）技术得到非线性函数的多项式近似表示，多项式的系数由不同阶次的偏导数所决定；这种方法的特点是展开点只有一个，其适用的前提是非线性函数具有显式的表达式且偏导数存在。另一种方法则是利用插值计算实现对非线性函数的近似，与 TSE 方法不同的是，这种方法用差分代替了微分，不需要偏导数运算，同时插值点不局限于单个点。

本章主要分析和探讨基于函数近似的高斯滤波方法，首先利用 TSE 法对随机变量经过一般非线性系统的统计特性进行了分析，在此基础上针对一阶 TSE 这一重要特例，导出了扩展卡尔曼滤波（EKF）算法，分析了其近似误差，并针对特定的应用条件介绍了常用的 EKF 衍生算法。另外，作为一种重要的插值近似方法，本章还重点阐述了二阶 Sterling 插值原理及基于这一插值原理的中心差分卡尔曼滤波（CDKF）算法，并给出了其求根实现形式。本章最后给出了两个目标跟踪的应用实例。

3.1　扩展卡尔曼滤波

3.1.1　随机变量的非线性传递

随机变量 $\boldsymbol{x}\in\mathbb{R}^L$ 满足 $\boldsymbol{x}\sim\mathcal{N}(\bar{\boldsymbol{x}},\boldsymbol{P}_x)$，经非线性系统 $\boldsymbol{y}=g(\boldsymbol{x})$ 传递后有

$$\boldsymbol{y}=g(\boldsymbol{x})=g(\bar{\boldsymbol{x}}+\boldsymbol{e}),\quad \boldsymbol{e}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{P}_x) \tag{3-1}$$

对上式在 $\bar{\boldsymbol{x}}$ 处进行多维泰勒展开，得

$$\boldsymbol{y}=g(\bar{\boldsymbol{x}})+D_{\boldsymbol{e}}g+\frac{1}{2!}D_{\boldsymbol{e}}^2g+\frac{1}{3!}D_{\boldsymbol{e}}^3g+\frac{1}{4!}D_{\boldsymbol{e}}^4g+\cdots \tag{3-2}$$

式中：$D_{\boldsymbol{e}}g$ 为全微分算子，表示在标称值 $\bar{\boldsymbol{x}}$ 处展开，并存在误差 $\boldsymbol{e}$。泰勒级数的第 i 项可表示为

$$\frac{1}{i!}D_{\boldsymbol{e}}^ig\triangleq\frac{1}{i!}\left(\sum_{j=1}^{L}\boldsymbol{e}_j\frac{\partial}{\partial x_j}\right)^i g(\boldsymbol{x})\Big|_{\boldsymbol{x}=\bar{\boldsymbol{x}}} \tag{3-3}$$

式中：$\boldsymbol{e}_j$ 为 $\boldsymbol{e}$ 的第 j 个分量。因此，第 i 项级数是关于 $\boldsymbol{e}$ 的 i 阶多项式，对其取期望，得

$$E\left[\frac{1}{i!}D_e^i g\right] \triangleq E\left[\frac{1}{i!}\left(\sum_{j=1}^{L} e_j \frac{\partial}{\partial x_j}\right)^i g(\boldsymbol{x})\big|_{\boldsymbol{x}=\bar{\boldsymbol{x}}}\right] = \frac{1}{i!}$$

$$\sum\nolimits_{k=1}^{L^i}\left[\boldsymbol{m}_{p(e,k,L,i)} \frac{\partial^i g(\boldsymbol{x})}{p_{sel}(\partial x,k,L,i)}\big|_{\boldsymbol{x}=\bar{\boldsymbol{x}}}\right] \tag{3-4}$$

式中：$p_{sel}(a,k,L,i)$ 为取 $\left(\sum_{j=1}^{L} a_j\right)^i$ 完全展开式的第 k 项，如

$$p_{sel}(\partial x,3,2,2) = \text{取}\left(\partial x_1 + \partial x_2\right)^2 \text{展开式的第 3 项} = \partial x_2 \partial x_1 \tag{3-5}$$

$\boldsymbol{m}_{e_1 e_2 \cdots e_n}$ 为 $\boldsymbol{e}$ 的 n 阶中心矩，即

$$\boldsymbol{m}_{e_1 e_2 \cdots e_n} = E\left[e_1 e_2 \cdots e_n\right] = \int\left(e_1 e_2 \cdots e_n\right) p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{3-6}$$

于是，$\boldsymbol{y}$ 的期望可表示为

$$\begin{aligned}\bar{\boldsymbol{y}} &= E[g(\bar{\boldsymbol{x}}+\boldsymbol{e})] = g(\bar{\boldsymbol{x}}) + \frac{1}{2}E[D_e^2 g] + E\left[\frac{1}{4!}D_e^4 g + \frac{1}{6!}D_e^6 g + \cdots\right] \\ &= g(\bar{\boldsymbol{x}}) + \frac{1}{2}\left[(\nabla^{\mathrm{T}} \boldsymbol{P}_x \nabla) g(\boldsymbol{x})\big|_{\boldsymbol{x}=\bar{\boldsymbol{x}}}\right] + E\left[\frac{1}{4!}D_e^4 g + \frac{1}{6!}D_e^6 g + \cdots\right]\end{aligned} \tag{3-7}$$

$$\begin{aligned}\boldsymbol{y} - \bar{\boldsymbol{y}} &= g(\bar{\boldsymbol{x}}+\boldsymbol{e}) - E[g(\bar{\boldsymbol{x}}+\boldsymbol{e})] \\ &= (D_e g + \frac{1}{2!}D_e^2 g + \frac{1}{3!}D_e^3 g + \cdots) - E\left[D_e g + \frac{1}{2!}D_e^2 g + \frac{1}{3!}D_e^3 g + \cdots\right]\end{aligned} \tag{3-8}$$

对式（3-8）取外积，并假定误差是对称分布的（即奇数阶矩为零），则有

$$\boldsymbol{P}_y = E[(\boldsymbol{y}-\bar{\boldsymbol{y}})(\boldsymbol{y}-\bar{\boldsymbol{y}})^{\mathrm{T}}] = \boldsymbol{G}_{\bar{x}} \boldsymbol{P}_x \boldsymbol{G}_{\bar{x}}^{\mathrm{T}} - \frac{1}{4}E[D_e^2 g]E[D_e^2 g]^{\mathrm{T}}$$

$$+E\left[\underbrace{\sum\nolimits_{i=1}^{\infty}\sum\nolimits_{j=1}^{\infty}\frac{1}{i!j!}D_e^i g\left(D_e^j g\right)^{\mathrm{T}}}_{\forall i,j:\ ij>1}\right] - \left[\underbrace{\sum\nolimits_{i=1}^{\infty}\sum\nolimits_{j=1}^{\infty}\frac{1}{(2i)!(2j)!}E\left[D_e^{2i} g\right]E\left[D_e^{2j} g\right]^{\mathrm{T}}}_{\forall i,j:\ ij>1}\right] \tag{3-9}$$

式中

$$\boldsymbol{G}_{\bar{x}} = \nabla_x g(\boldsymbol{x})\big|_{\boldsymbol{x}=\bar{\boldsymbol{x}}}$$

若对泰勒展开取一阶线性化近似（即忽略高阶项的影响），则

$$\boldsymbol{y} \approx \boldsymbol{y}^{lin} = g(\bar{\boldsymbol{x}}) + D_e g \tag{3-10}$$

$$\bar{\boldsymbol{y}} \approx \bar{\boldsymbol{y}}^{lin} = g(\bar{\boldsymbol{x}}) \tag{3-11}$$

$$\boldsymbol{P}_y \approx \boldsymbol{P}_y^{lin} = \boldsymbol{G}_{\bar{x}} \boldsymbol{P}_x \boldsymbol{G}_{\bar{x}}^{\mathrm{T}} \tag{3-12}$$

3.1.2 扩展卡尔曼滤波算法

下面将在 3.1.1 节分析结果的基础上给出 EKF 算法的流程，并通过一个简单的算例分析 EKF 由线性化所引入的误差。在此基础上，讨论 EKF 的几种衍生算法，并指出其可能存在的优点和不足。

仍然考虑下列方程所描述的非线性系统

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}) + \boldsymbol{w}_k \tag{3-13}$$

$$\boldsymbol{y}_k = h(\boldsymbol{x}_k) + \boldsymbol{v}_k \tag{3-14}$$

式中：$f(\cdot)$和$h(\cdot)$为一般非线性函数；$\boldsymbol{w}_k$和$\boldsymbol{v}_k$分别为零均值过程噪声和测量噪声。

由 1.2 节中的分析可知，当满足以下两个条件时：

（1）$\boldsymbol{w}_k$和$\boldsymbol{v}_k$为相互独立的高斯噪声，其协方差分别为$\boldsymbol{Q}_k$和$\boldsymbol{R}_k$；

（2）$f(\boldsymbol{x}_{k-1})$为$\boldsymbol{x}_{k-1}$的线性函数，$h(\boldsymbol{x}_k)$为$\boldsymbol{x}_k$的线性函数。

递推贝叶斯滤波公式存在解析解，相应地得到的滤波算法为线性、高斯条件下的最优卡尔曼滤波算法。对于一般的非线性方程，式（3-13）和式（3-14）不能简单写成线性表示形式，后验分布的解析表达式也就难以求得。此时，一种可行的方案是对非线性函数$f(\cdot)$、$h(\cdot)$进行局部线性化，通常写成一阶泰勒展开的形式，即

$$\begin{aligned} f(\boldsymbol{x}_{k-1}) &= f(\hat{\boldsymbol{x}}_{k-1}) + \left.\frac{\partial f(\boldsymbol{x}_{k-1})}{\partial \boldsymbol{x}_{k-1}}\right|_{x_{k-1}=\hat{x}_{k-1}} (\boldsymbol{x}_{k-1} - \hat{\boldsymbol{x}}_{k-1}) + H.O.T \\ &\approx f(\hat{\boldsymbol{x}}_{k-1}) + \boldsymbol{F}_k \tilde{\boldsymbol{x}}_{k-1} \end{aligned}$$

$$\begin{aligned} h(\boldsymbol{x}_k) &= h(\hat{\boldsymbol{x}}_{k|k-1}) + \left.\frac{\partial h(\boldsymbol{x}_k)}{\partial \boldsymbol{x}_k}\right|_{x_{k-1}=\hat{x}_{k|k-1}} (\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1}) + H.O.T \\ &\approx h(\hat{\boldsymbol{x}}_{k|k-1}) + \boldsymbol{H}_k \tilde{\boldsymbol{x}}_{k|k-1} \end{aligned}$$

并将$p(\boldsymbol{x}_{k-1} \mid \boldsymbol{y}_{1:k-1})$、$p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1})$和$p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k})$近似为高斯分布，即

$$p(\boldsymbol{x}_{k-1} \mid \boldsymbol{y}_{1:k-1}) \approx \mathcal{N}(\boldsymbol{x}_{k-1}; \hat{\boldsymbol{x}}_{k-1}, \boldsymbol{P}_{k-1}) \tag{3-15}$$

$$p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1}) \approx \mathcal{N}(\boldsymbol{x}_k; \hat{\boldsymbol{x}}_{k|k-1}, \boldsymbol{P}_{k|k-1}) \tag{3-16}$$

$$p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k}) \approx \mathcal{N}(\boldsymbol{x}_k; \hat{\boldsymbol{x}}_k, \boldsymbol{P}_k) \tag{3-17}$$

其中

$$\hat{\boldsymbol{x}}_{k|k-1} = f(\hat{\boldsymbol{x}}_{k-1}) \tag{3-18}$$

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{F}_k \boldsymbol{P}_{k-1} \boldsymbol{F}_k^{\mathrm{T}} + \boldsymbol{Q}_k \tag{3-19}$$

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k [\boldsymbol{y}_k - h(\hat{\boldsymbol{x}}_{k|k-1})] \tag{3-20}$$

$$\boldsymbol{P}_k = \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k \boldsymbol{H}_k \boldsymbol{P}_{k|k-1} \tag{3-21}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^{\mathrm{T}} (\boldsymbol{H}_k \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^{\mathrm{T}} + \boldsymbol{R}_k)^{-1} \tag{3-22}$$

$$\boldsymbol{F}_k = \left.\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{x=\hat{x}_{k-1}}, \quad \boldsymbol{H}_k = \left.\frac{\partial h(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{x=\hat{x}_{k|k-1}} \tag{3-23}$$

式（3-18）～式（3-23）构成了扩展卡尔曼滤波（EKF）的递推过程，其中$\boldsymbol{F}_k$和$\boldsymbol{H}_k$为多维状态泰勒展开得到的 Jacobian 矩阵。完整的 EKF 算法[1]流程可概括如下。

Alg（3.1）：Extended Kalman Filtering

步骤 1：初始化

给定初始条件$\hat{\boldsymbol{x}}_0$，$\boldsymbol{P}_0$，$\boldsymbol{Q}_0$，$\boldsymbol{R}_0$。

步骤 2：预测

（1）计算状态模型式（3-13）的 Jacobian 矩阵，即

$$\boldsymbol{F}_k = \nabla_x f(\boldsymbol{x})\big|_{x=\hat{x}_{k-1}} \tag{3-24}$$

（2）时间更新，即

$$\hat{\boldsymbol{x}}_{k|k-1} = f(\hat{\boldsymbol{x}}_{k-1}) \tag{3-25}$$

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{F}_k \boldsymbol{P}_{k-1} \boldsymbol{F}_k^{\mathrm{T}} + \boldsymbol{Q}_k \tag{3-26}$$

步骤 3：修正

（1）计算观测模型式（3-14）的 Jacobian 矩阵，即

$$\boldsymbol{H}_k = \nabla_{\boldsymbol{x}} h(\boldsymbol{x})\big|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{k|k-1}} \tag{3-27}$$

（2）测量更新，即

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^{\mathrm{T}} (\boldsymbol{H}_k \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^{\mathrm{T}} + \boldsymbol{R}_k)^{-1} \tag{3-28}$$

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k [\boldsymbol{y}_k - h(\hat{\boldsymbol{x}}_{k|k-1})] \tag{3-29}$$

$$\boldsymbol{P}_k = \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k \boldsymbol{H}_k \boldsymbol{P}_{k|k-1} \tag{3-30}$$

3.2 EKF 的误差分析与改进方法

3.2.1 EKF 的近似误差分析

考虑下列简单的随机变量非线性传递过程

$$y = g(x) = ax^2 + b, \quad x \sim \mathcal{N}(x; \bar{x}, \sigma_x^2) \tag{3-31}$$

式中：a、b 为常数。

记求导算式 $g^{(n)}(\bar{x}) = g^{(n)}(x)\big|_{x=\bar{x}}$，则根据式（3-7）～式（3-9），得

$$\begin{aligned} \bar{y} &= g(\bar{x}) + E\left[(x-\bar{x})g^{(1)}(\bar{x}) + \frac{1}{2}(x-\bar{x})^2 g^{(2)}(\bar{x})\right] \\ &= a\bar{x}^2 + E\left[2a(x-\bar{x})\bar{x} + a(x-\bar{x})^2\right] + b = a\bar{x}^2 + a\sigma_x^2 + b = a(\bar{x}^2 + \sigma_x^2) + b \end{aligned} \tag{3-32}$$

$$\begin{aligned} \sigma_y^2 &= (2a\bar{x})^2 \sigma_x^2 - \frac{1}{4} E^2[2a(x-\bar{x})^2] + E[4a^2\bar{x}(x-\bar{x})^3] + E[a^2(x-\bar{x})^4] \\ &= 4a^2\bar{x}^2\sigma_x^2 - a^2(\sigma_x^2)^2 + 4a^2\bar{x}E[(x-\bar{x})^3] + a^2 E[(x-\bar{x})^4] \end{aligned} \tag{3-33}$$

由于 x 为高斯随机变量，故 $E[(x-\bar{x})^3] = 0$，$E[(x-\bar{x})^4] = 3(\sigma_x^2)^2$，于是有

$$\sigma_y^2 = 2a^2\sigma_x^4 + 4a^2\bar{x}^2\sigma_x^2 \tag{3-34}$$

显然，取一阶线性化所得的均值和方差分别为

$$\bar{y}^{lin} = g(\bar{x}) = a\bar{x}^2 + b \tag{3-35}$$

$$(\sigma_y^2)^{lin} = g^{(2)}(\bar{x})\sigma_x^2 g^{(2)}(\bar{x}) = 4a^2\bar{x}^2\sigma_x^2 \tag{3-36}$$

为了更好地说明线性化方法对近似精度的影响，假定 $a=2$，$b=1$，$\bar{x}=2$，表 3.1 对不同方差 σ_x^2 条件下的近似性能进行了比较。在表中，蒙特卡罗近似法采用的样本数为 200000。

表 3.1　线性化与蒙特卡罗方法近似精度比较

$\bar{x}=1$	$\sigma_x^2=0.1$		$\sigma_x^2=1$		$\sigma_x^2=2$	
	$\bar{y}$	σ_y^2	$\bar{y}$	σ_y^2	$\bar{y}$	σ_y^2
Analytical	9.2000	6.4800	11.0000	72.0000	13.0000	160.000
蒙特卡罗	9.2032	6.4828	10.9576	72.1043	12.9658	158.658
EKF	9.0000	6.4000	9.0000	64.0000	9.0000	128.000

由表 3.1 中的结果可以看出，当初始误差较小时，线性化方法所得的均值和方差都比较接近真实值。这说明，初始误差较小的情况下，局部线性化是合理的。随着初始误差的增加，线性化结果（特别是方差）与真值间的差距越来越大，这就是 EKF 算法对初值敏感、滤波器性能不稳定的本质原因。EKF 的近似误差也可用图 3.1 来直观地说明，由图中可以看出，由于线性化过程存在较大的误差，致使后验均值和协方差估计不准确。

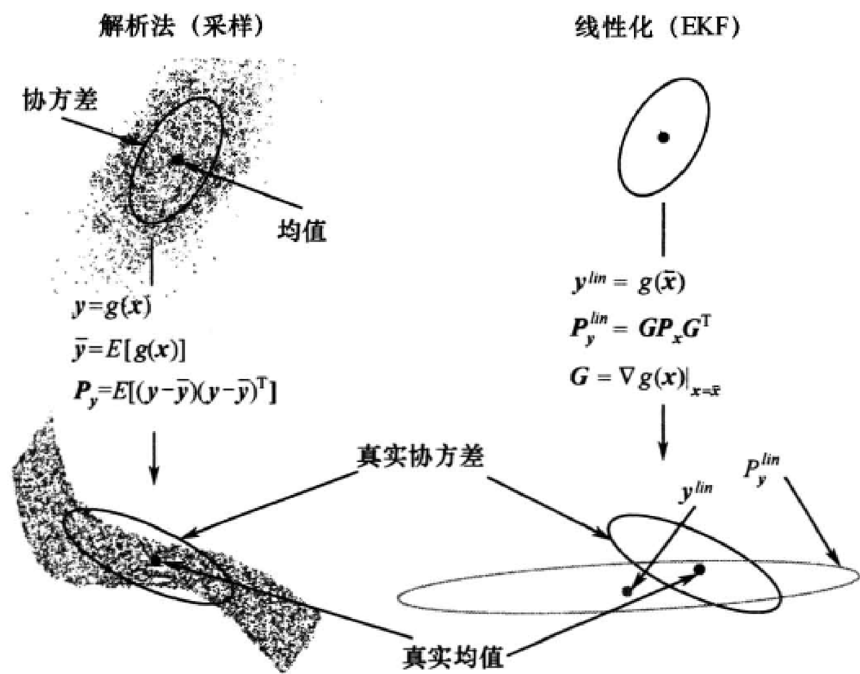


图 3.1　线性化近似误差表示

3.2.2　EKF 的几种改进方法

1. 二阶 **EKF**（泰勒展开在二阶以上截断）

由前面的分析可知，只有在满足局部线性化（即标称状态或展开点与真实状态充分接近）的条件下，EKF 才能达到较好的效果。为了提高估计性能，对 $f(\boldsymbol{x}_{k-1})$ 和 $h(\boldsymbol{x}_k)$ 分别在 $\hat{\boldsymbol{x}}_{k-1}$ 和 $\hat{\boldsymbol{x}}_{k|k-1}$ 处进行泰勒展开，并取展开式的前三项，即可推导出非线性系统的二阶滤波公式。当然，在展开式中可以保留前四项或五项。但实践结果表明，二阶 EKF 的性能要好于 EKF；而三阶、四阶滤波与二阶滤波相比，性能改善并不明显，且在状态量维数较高时运算相当复杂，故一般很少采用。

很明显，二阶滤波器需要求 Hessian 矩阵，这将使计算量大大增加，不利于实时应用。一般情况下，只有当系统非线性程度较高或系统状态为标量的情况下才考虑用二阶 EKF。

2. IEKF（迭代 EKF）

IEKF 的基本思想是，在获得 k 时刻的估计值 $\hat{\boldsymbol{x}}_k$ 和估计协方差 $\boldsymbol{P}_k$ 以后，将其代替预测值 $\hat{\boldsymbol{x}}_{k|k-1}$ 和预测误差协方差 $\boldsymbol{P}_{k|k-1}$ 再进行 EKF 处理[2]，即

$$\hat{\boldsymbol{x}}_{k,0}=\hat{\boldsymbol{x}}_{k|k-1}\text{，}\quad \boldsymbol{P}_{k,0}=\boldsymbol{P}_{k|k-1} \tag{3-37}$$

$$\hat{\boldsymbol{x}}_{k,j+1}=\hat{\boldsymbol{x}}_{k|k-1}+\boldsymbol{K}_{k,j}[\boldsymbol{y}_k-h(\hat{\boldsymbol{x}}_{k,j})-\boldsymbol{H}_{k,j}(\hat{\boldsymbol{x}}_{k|k-1}-\hat{\boldsymbol{x}}_{k,j})] \tag{3-38}$$

$$\boldsymbol{P}_{k,j}=(\boldsymbol{I}-\boldsymbol{K}_{k,j}\boldsymbol{H}_{k,j})\boldsymbol{P}_{k|k-1} \tag{3-39}$$

式中

$$\boldsymbol{H}_{k,j}=\left.\frac{\partial h(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{x=\hat{x}_{k,j}}\text{，}\quad \boldsymbol{K}_{k,j}=\boldsymbol{P}_{k|k-1}\boldsymbol{H}_{k,j}^{\mathrm{T}}(\boldsymbol{H}_{k,j}\boldsymbol{P}_{k|k-1}\boldsymbol{H}_{k,j}^{\mathrm{T}}+\boldsymbol{R}_k)^{-1} \tag{3-40}$$

重复式（3-38）～式（3-40）直到满足条件 $\left\|\hat{\boldsymbol{x}}_{k,j+1}-\hat{\boldsymbol{x}}_{k,j}\right\|\leqslant V_{th}$（$V_{th}$ 为预设的门限），这样若总共进行 $j=1,2,\cdots,N$ 次迭代，最终输出结果为 $\hat{\boldsymbol{x}}_k=\hat{\boldsymbol{x}}_{k,N}, \boldsymbol{P}_k=\boldsymbol{P}_{k,N}$。

3. MGEKF（修正增益 EKF）

若测量方程 $\boldsymbol{y}_k=h(\boldsymbol{x}_k)$ 满足

$$h(\boldsymbol{x}_k)-h(\hat{\boldsymbol{x}}_{k|k-1})=\boldsymbol{G}_k(\boldsymbol{y}_k,\hat{\boldsymbol{x}}_{k|k-1})(\boldsymbol{x}_k-\hat{\boldsymbol{x}}_{k|k-1}) \tag{3-41}$$

也就是说这个非线性函数 $\boldsymbol{y}_k=h(\boldsymbol{x}_k)$ 可以写成状态 $\boldsymbol{x}_k$ 的线性形式，则称该非线性函数是可修正的[3,4]，同时也就可以应用 MGEKF 滤波算法。此时，滤波估计误差的协方差矩阵为

$$\boldsymbol{P}_k=[\boldsymbol{I}-\boldsymbol{K}_k\boldsymbol{G}_k(\boldsymbol{y}_{mk},\hat{\boldsymbol{x}}_{k|k-1})]\boldsymbol{P}_{k|k-1}[\boldsymbol{I}-\boldsymbol{K}_k\boldsymbol{G}_k(\boldsymbol{y}_{mk},\hat{\boldsymbol{x}}_{k|k-1})]^{\mathrm{T}}+\boldsymbol{K}_k\boldsymbol{R}_k\boldsymbol{K}_k^{\mathrm{T}} \tag{3-42}$$

式中采用 $\boldsymbol{y}_{mk}$ 代替 $\boldsymbol{y}_k$ 是因为实际中只能得到带有噪声的测量值，其他计算公式与 EKF 算法一致。

对于不同的系统，MGEKF 算法具有不同的变换矩阵 $\boldsymbol{G}_k(\boldsymbol{y}_{mk},\hat{\boldsymbol{x}}_k)$，推导相应系统条件下的 $\boldsymbol{G}_k(\boldsymbol{y}_{mk},\hat{\boldsymbol{x}}_k)$ 是 MGEKF 算法的关键。在二维平面 (x_k,y_k) 只测角条件下，有

$$\varphi_k=h(\boldsymbol{x}_k)=\arctan(x_k/y_k) \tag{3-43}$$

此时有

$$\boldsymbol{G}_k=\begin{bmatrix}0 & 0 & \dfrac{\cos\varphi_{mk}}{\hat{r}_k} & \dfrac{-\sin\varphi_{mk}}{\hat{r}_k}\end{bmatrix} \tag{3-44}$$

在三维空间 (x_k,y_k,z_k) 只测角条件下，有

$$\begin{bmatrix}\varphi_k\\ \theta_k\end{bmatrix}=h(\boldsymbol{x}_k)=\begin{bmatrix}\arctan(x_k/y_k)\\ \arctan\left(\sqrt{x_k^2+y_k^2}/z_k\right)\end{bmatrix} \tag{3-45}$$

相应地，可得到

$$\boldsymbol{G}_k=\begin{bmatrix}\dfrac{\cos\varphi_{mk}}{\hat{r}_k} & \dfrac{-\sin\varphi_{mk}}{\hat{r}_k} & 0 & 0 & 0 & 0\\ \dfrac{\cos\hat{\theta}_k\sin[(\varphi_{mk}+\hat{\varphi}_k)/2]}{\hat{\rho}_k\cos[(\varphi_{mk}-\hat{\varphi}_k)/2]} & \dfrac{\cos\hat{\theta}_k\cos[(\varphi_{mk}+\hat{\varphi}_k)/2]}{\hat{\rho}_k\cos[(\varphi_{mk}-\hat{\varphi}_k)/2]} & \dfrac{-\sin\hat{\theta}_k}{\hat{\rho}_k} & 0 & 0 & 0\end{bmatrix} \tag{3-46}$$

式中

$\hat{r}_k = \hat{x}_{k|k-1}\sin\beta_{mk} + \hat{y}_{k|k-1}\cos\beta_{mk}$, $\hat{\rho}_k = \sqrt{\hat{r}_k^2 + (\hat{z}_{k|k-1})^2} = \hat{r}_k \sin\theta_{mk} + \hat{z}_{k|k-1}\cos\theta_{mk}$, $\hat{\varphi}_k = \arctan(\hat{x}_{k|k-1} / \hat{y}_{k|k-1})$。

尽管 MGEKF 在性能上比 EKF 有了提高，但对于一般的观测方程，修正矩阵不存在或很难求得，这限制了该算法的应用。

4. MVEKF（修正协方差 EKF）

在这种算法中[5]，认为状态滤波估计 $\hat{\boldsymbol{x}}_k$ 比状态预测 $\hat{\boldsymbol{x}}_{k|k-1}$ 得到的结果更为精确，从而利用 $\hat{\boldsymbol{x}}_k$ 来计算 Jacobian 矩阵 $\boldsymbol{H}_k^+ = \left.\dfrac{\partial h(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_k}$；而后再利用该矩阵再计算一次滤波误差的协方差矩阵，即

$$\boldsymbol{K}_k^+ = \boldsymbol{P}_{k|k-1}(\boldsymbol{H}_k^+)^{\mathrm{T}}(\boldsymbol{H}_k^+\boldsymbol{P}_{k|k-1}(\boldsymbol{H}_k^+)^{\mathrm{T}} + \boldsymbol{R}_k)^{-1}$$

$$\boldsymbol{P}_k^+ = (\boldsymbol{I} - \boldsymbol{K}_k^+\boldsymbol{H}_k^+)\boldsymbol{P}_{k|k-1}$$

上述过程可认为是对传统滤波误差协方差矩阵的修正。仔细比较 MVEKF 和迭代次数为 2 的 IEKF 算法不难发现，两者在本质是一致的，即通过多次修正提高估计精度；所不同的是，MVEKF 比 IEKF 少了一次更新状态估计的环节。

表 3.2 对 EKF 几种衍生算法的特点作了简单的比较，实际应用中可根据需要对不同的算法进行合理选择。

表 3.2　几种 EKF 衍生算法特点比较

二阶 EKF		IEKF		MGEKF	
优点	缺点	优点	缺点	优点	缺点
近似精度比 EKF 有了较大提高	运算开销随状态维数增加显著增大	在保证观测精度的前提下可加速收敛速度	必须以高精度观测为前提，运算量也有所提高	可改善协方差矩阵特性，提高算法稳定性	修正矩阵难推导或不存在

3.3　中心差分卡尔曼滤波

在前文介绍的 EKF 及其衍生算法中，尽管针对各种不同的应用做了适当的修正，但其理论基础是相同的，即用 TSE 去近似非线性系统函数。这类方法的共同特点是非线性函数的近似采用的是单点近似，且通常只取一阶逼近，因此近似精度受到很大限制，在系统非线性程度较高的情况下更是如此。

对非线性函数近似的另一种有效方式是采用插值公式，与采用解析求导的方法不同，插值公式利用了若干个插值点的函数值来计算。根据不同的需要，可能有不同的插值公式，其中应用最广泛的一种插值方法是 Sterling。接下来将对这种插值方法及其由此导出的卡尔曼滤波算法进行详细介绍。

3.3.1　二阶 Sterling 多项式插值

在文献[6]中利用了中心分开差分（Central Divided Difference，CDD）内插公式，称为 Sterling 多项式内插公式。对于二阶标量情况，该内插公式可记为

$$g(x)=g(\overline{x})+D_{\Delta x}g+\frac{1}{2!}D_{\Delta x}^{2}g \tag{3-47}$$

式中：$D_{\Delta x}$、$D_{\Delta x}^{2}$ 分别为作用于函数 $g(x)$ 的一阶和二阶中心分开差分（CDD）算子。

对于标量情况

$$D_{\Delta x}g=(x-\overline{x})\frac{g(\overline{x}+h)-g(\overline{x}-h)}{2h} \tag{3-48}$$

$$D_{\Delta x}^{2}g=(x-\overline{x})^{2}\frac{g(\overline{x}+h)+g(\overline{x}-h)-2g(\overline{x})}{h^{2}} \tag{3-49}$$

式中：h 为中心差分步长；$\overline{x}$ 为展开点的先验均值。通过类比可知，Sterling 内插公式可认为是用 CDD 方法代替泰勒级数展开法中的解析求导。

对于多维状态量，$g(\boldsymbol{x})$ 可通过对先验变量 $\boldsymbol{x}\in\mathbb{R}^{L}$ 的“统计解耦”处理得到，其处理过程可利用下面的线性变换来实现，即

$$\boldsymbol{z}=\boldsymbol{S}_{x}^{-1}\boldsymbol{x} \tag{3-50}$$

$$\tilde{g}(\boldsymbol{z})\approx g(\boldsymbol{S}_{x}\boldsymbol{z})=g(\boldsymbol{x}) \tag{3-51}$$

式中：$\boldsymbol{S}_{x}$ 是变量 $\boldsymbol{x}$ 协方差矩阵 $\boldsymbol{P}_{x}$ 的 Cholesky 因子，即

$$\boldsymbol{P}_{x}=\boldsymbol{S}_{x}\boldsymbol{S}_{x}^{\mathrm{T}} \tag{3-52}$$

式（3-52）中假定了随机变量 $\boldsymbol{x}$ 具有均值 $\overline{\boldsymbol{x}}=E[\boldsymbol{x}]$、方差 $\boldsymbol{P}_{x}=E[(\boldsymbol{x}-\overline{\boldsymbol{x}})(\boldsymbol{x}-\overline{\boldsymbol{x}})^{\mathrm{T}}]$。注意到 $\tilde{g}(\cdot)$ 和 $g(\cdot)$ 的泰勒展开是等效的，上述变换实现了对 $\boldsymbol{x}$ 的统计解耦，这是因为经变换后 $\boldsymbol{z}$ 的各分量变为互不相关的量（具有单位方差），即

$$\boldsymbol{P}_{z}=E[(\boldsymbol{z}-\overline{\boldsymbol{z}})(\boldsymbol{z}-\overline{\boldsymbol{z}})^{\mathrm{T}}]=\boldsymbol{I} \tag{3-53}$$

上述处理过程使得标量中心差分算子式（3-48）和式（3-49）能够独立地应用于 $\tilde{g}(\boldsymbol{z})$ 的各个分量，从而得到多维状态量的中心差分算子为

$$\tilde{D}_{\Delta_{z}}\tilde{g}=(\sum_{i=1}^{L}\Delta_{z_i}m_i d_i)\tilde{g}(\overline{\boldsymbol{z}}) \tag{3-54}$$

$$\tilde{D}_{\Delta_{z}}^{2}\tilde{g}=[\sum_{i=1}^{L}\Delta_{z_i}^{2}d_i^{2}+\sum_{j=1}^{L}\sum_{\substack{q=1\\q\neq j}}^{L}\Delta_{z_j}\Delta_{z_q}(m_j d_j)(m_q d_q)]\tilde{g}(\overline{\boldsymbol{z}}) \tag{3-55}$$

式中：$\Delta_{z_i}=(\boldsymbol{z}-\overline{\boldsymbol{z}})_i\ (i=1,2,\cdots,L)$ 为 $\boldsymbol{z}-\overline{\boldsymbol{z}}$ 的第 i 个分量；L 为 $\boldsymbol{z}$ 的维数；m_i、d_i、d_i^2 分别为均值、一阶差分和二阶差分算子，其定义为

$$m_i\tilde{g}(\overline{\boldsymbol{z}})=\frac{1}{2}[\tilde{g}(\overline{\boldsymbol{z}}+h\boldsymbol{e}_i)+\tilde{g}(\overline{\boldsymbol{z}}-h\boldsymbol{e}_i)] \tag{3-56}$$

$$d_i\tilde{g}(\overline{\boldsymbol{z}})=\frac{1}{2h}[\tilde{g}(\overline{\boldsymbol{z}}+h\boldsymbol{e}_i)-\tilde{g}(\overline{\boldsymbol{z}}-h\boldsymbol{e}_i)] \tag{3-57}$$

$$d_i^2\tilde{g}(\overline{\boldsymbol{z}})=\frac{1}{2h^2}[\tilde{g}(\overline{\boldsymbol{z}}+h\boldsymbol{e}_i)+\tilde{g}(\overline{\boldsymbol{z}}-h\boldsymbol{e}_i)-2\tilde{g}(\overline{\boldsymbol{z}})] \tag{3-58}$$

式中：$\boldsymbol{e}_i$ 为第 i 个单位矢量。

根据式（3-50）和式（3-51），易得

$$\tilde{g}(\overline{\boldsymbol{z}}\pm h\boldsymbol{e}_i)=g(\boldsymbol{S}_x[\overline{\boldsymbol{z}}\pm h\boldsymbol{e}_i])=g(\boldsymbol{S}_x\overline{\boldsymbol{z}}\pm h\boldsymbol{S}_x\boldsymbol{e}_i])=g(\overline{\boldsymbol{x}}\pm h\boldsymbol{s}_{xi}]) \tag{3-59}$$

式中：$\boldsymbol{s}_{xi}$ 为协方差矩阵 Cholesky 分解因子的第 i 列，即

$$s_{xi}=S_x e_i=(S_x)_i=\left(\sqrt{P_x}\right)_i \tag{3-60}$$

3.3.2 插值后的均值和方差计算

不失一般性，仍假设具有 $\bar{x}$ 均值、P_x 方差的 L 维状态量 x 经任意非线性函数 $g(\cdot)$ 传递问题，即

$$y=g(x) \tag{3-61}$$

对式（3-61）利用多维变量的二阶 Sterling 插值公式进行展开，得

$$y=g(x)=\tilde{g}(z)\approx\tilde{g}(\bar{z})+D_{\Delta_z}\tilde{g}+\frac{1}{2}D_{\Delta_z}^2\tilde{g} \tag{3-62}$$

式中：$\bar{z}=S_x\bar{x}$。

于是 y 的均值、协方差、互协方差可表示为

$$\bar{y}=E[y] \tag{3-63}$$

$$P_y=E[(y-\bar{y})(y-\bar{y})^{\mathrm{T}}] \tag{3-64}$$

$$P_{xy}=E[(x-\bar{x})(y-\bar{y})^{\mathrm{T}}] \tag{3-65}$$

根据式（3-62）的 Sterling 插值，可得均值 $\bar{y}$ 的近似值为

$$\begin{aligned}\bar{y}&\approx E[\tilde{g}(\bar{z})+D_{\Delta_z}\tilde{g}+\frac{1}{2}D_{\Delta_z}^2\tilde{g}]=\tilde{g}(\bar{z})+E[\frac{1}{2}D_{\Delta_z}^2\tilde{g}]\\&=\tilde{g}(\bar{z})+E[\frac{1}{2h^2}(\sum_{i=1}^{L}\Delta_{z_i}^2\mathrm{d}_i^2)\tilde{g}(\bar{z})]\\&=\tilde{g}(\bar{z})+\frac{1}{2h^2}\sum_{i=1}^{L}[\tilde{g}(\bar{z}+he_i)+\tilde{g}(\bar{z}-he_i)-2\tilde{g}(\bar{z})]\\&=\frac{h^2-L}{h^2}\tilde{g}(\bar{z})+\frac{1}{2h^2}\sum_{i=1}^{L}[\tilde{g}(\bar{z}+he_i)+\tilde{g}(\bar{z}-he_i)]\end{aligned} \tag{3-66}$$

式中，假定 $\Delta_z=z-\bar{z}$ 为零均值、单位协方差随机变量。

将式（3-59）代入式（3-66）可得用 $\bar{x}$ 表示的均值为

$$\bar{y}\approx\frac{h^2-L}{h^2}g(\bar{x})+\frac{1}{2h^2}\sum_{i=1}^{L}[g(\bar{x}+hs_{x_i})+g(\bar{x}-hs_{x_i})] \tag{3-67}$$

计算协方差矩阵的近似值时，可采用式（3-64）的展开式，即

$$\begin{aligned}P_y&=E[(y-\bar{y})(y-\bar{y})^{\mathrm{T}}]\\&=E[(y-g(\bar{x}))(y-g(\bar{x}))^{\mathrm{T}}]-E[y-g(\bar{x})]E[y-g(\bar{x})]^{\mathrm{T}}\\&=E[(y-\tilde{g}(\bar{z}))(y-\tilde{g}(\bar{z}))^{\mathrm{T}}]-E[y-\tilde{g}(\bar{z})]E[y-\tilde{g}(\bar{z})]^{\mathrm{T}}\end{aligned} \tag{3-68}$$

在计算上式时，利用了恒等式

$$\begin{aligned}\bar{y}&=E[y]=E[y]+g(\bar{x})-g(\bar{x})\\&=E[y]+g(\bar{x})-E[g(\bar{x})]=g(\bar{x})+E[y-g(\bar{x})]\end{aligned} \tag{3-69}$$

根据式（3-62），得

$$y-\tilde{g}(\bar{z})=D_{\Delta_z}\tilde{g}+\frac{1}{2}D_{\Delta_z}^2\tilde{g} \tag{3-70}$$

式（3-70）即为 $y-\tilde{g}(\bar{z})$ 的二阶 Sterling 插值近似。将式（3-70）代入到式（3-68），得

$$\begin{aligned}\boldsymbol{P}_y \approx & E[(D_{\Delta_z}\tilde{g}+\frac{1}{2}D_{\Delta_z}^2\tilde{g})(D_{\Delta_z}\tilde{g}+\frac{1}{2}D_{\Delta_z}^2\tilde{g})^{\mathrm{T}}] \\ & -E[D_{\Delta_z}\tilde{g}+\frac{1}{2}D_{\Delta_z}^2\tilde{g}]E[D_{\Delta_z}\tilde{g}+\frac{1}{2}D_{\Delta_z}^2\tilde{g}]^{\mathrm{T}}\end{aligned} \tag{3-71}$$

类似前面的处理，仍假定 $\Delta_z=z-\bar{z}$ 为零均值、单位协方差对称随机变量。由于具有对称性，所有奇数阶矩为零；同时，为了方便计算，将舍弃所有四阶项的分量 $E\left[\frac{1}{4}D_{\Delta_z}^2\tilde{g}(D_{\Delta_z}^2\tilde{g})^{\mathrm{T}}\right]$（该项包含了式（3-71）展开式中的互差分，忽略四阶项的目的是为了避免因 z 的维数增加导致的计算量的快速增加）。基于上述假定和处理，对式（3-71）进行展开并取展开式中的期望，可得协方差的近似值为

$$\begin{aligned}\boldsymbol{P}_y \approx & \frac{1}{4h^2}\sum_{i=1}^{L}[g(\bar{x}+hs_{x_i})-g(\bar{x}-hs_{x_i})] \\ & [g(\bar{x}+hs_{x_i})-g(\bar{x}-hs_{x_i})]^{\mathrm{T}} \\ & +\frac{h^2-1}{4h^4}\sum_{i=1}^{L}[g(\bar{x}+hs_{x_i})+g(\bar{x}-hs_{x_i})- \\ & 2g(\bar{x})][g(\bar{x}+hs_{x_i})+g(\bar{x}-hs_{x_i})-2g(\bar{x})]^{\mathrm{T}}\end{aligned} \tag{3-72}$$

类似地，可得到互协方差阵的近似值为

$$\begin{aligned}\boldsymbol{P}_{xy} &= E[(x-\bar{x})(y-\bar{y})^{\mathrm{T}}] \\ &\approx E[(\boldsymbol{S}_x(z-\bar{z}))(\tilde{D}_{\Delta_z}\tilde{g}+\frac{1}{2}\tilde{D}_{\Delta_z}^2\tilde{g}-E[\frac{1}{2}\tilde{D}_{\Delta_z}^2\tilde{g}])^{\mathrm{T}}] \\ &= E[(\boldsymbol{S}_x\Delta_z)(\tilde{D}_{\Delta_z}\tilde{g})^{\mathrm{T}}]+\frac{1}{2}E[(\boldsymbol{S}_x\Delta_z)(\tilde{D}_{\Delta_z}^2\tilde{g})^{\mathrm{T}}]-\frac{1}{2}E[(\boldsymbol{S}_x\Delta_z)]E[\frac{1}{2}\tilde{D}_{\Delta_z}^2\tilde{g}] \\ &= E[(\boldsymbol{S}_x\Delta_z)(\tilde{D}_{\Delta_z}\tilde{g})^{\mathrm{T}}] \\ &= \frac{1}{2h}\sum_{i=1}^{L}s_{x_i}[\tilde{g}(\bar{z}+he_i)-\tilde{g}(\bar{z}-he_i)]^{\mathrm{T}} \\ &= \frac{1}{2h}\sum_{i=1}^{L}s_{x_i}[g(\bar{x}+hs_{x_i})-g(\bar{x}-hs_{x_i})]^{\mathrm{T}}\end{aligned} \tag{3-73}$$

式中利用了 Δ_z 的对称性特点及其所对应的奇数阶矩为零的性质。

中心差分步长参数 h 的最优设置取决于 $z=\boldsymbol{S}_x^{-1}x$ 的先验分布，分析表明[7]，为了使泰勒级数展开得到的均值和方差与近似估计得到的结果之间误差最小，h^2 与 z 的峭度①应当相等。对于高斯先验，最优的 h 取值为 $\sqrt{3}$。

一旦获得随机变量经非线性变换后（经 Sterling 插值）得到的后验统计量，就可以很容易将这一结果应用于卡尔曼滤波算法框架中得到中心差分卡尔曼滤波。

---

① 峭度（Kurtosis）：其定义为 $k=E[x-\mu]^4/\sigma^4$，μ 为随机变量 x 的均值，σ 为标准差；该指标用于描述某种分布偏离高斯分布的程度，当 x 服从高斯分布时 k=3。

3.3.3 中心差分卡尔曼滤波器

中心差分卡尔曼滤波器是 Sterling 插值在递推卡尔曼滤波框架下的直接应用，用式（3-67）、式（3-72）和式（3-73）计算经插值近似后的后验均值、协方差及卡尔曼滤波增益，即可得到 CDKF 滤波算法，具体流程可概括如下。

Alg（3.2）：Central Difference Kalman Filter （CDKF）

步骤 1：初始化

$$\hat{\boldsymbol{x}}_0 = E[\boldsymbol{x}_0] \qquad \boldsymbol{P}_{\boldsymbol{x}_0} = E[(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)^{\mathrm{T}}]$$

$$\bar{\boldsymbol{w}} = E[\boldsymbol{w}] \qquad \boldsymbol{R}_{\boldsymbol{w}} = E[(\boldsymbol{w} - \bar{\boldsymbol{w}})(\boldsymbol{w} - \bar{\boldsymbol{w}})^{\mathrm{T}}]$$

$$\bar{\boldsymbol{v}} = E[\boldsymbol{v}] \qquad \boldsymbol{R}_{\boldsymbol{v}} = E[(\boldsymbol{v} - \hat{\boldsymbol{v}})(\boldsymbol{v} - \hat{\boldsymbol{v}})^{\mathrm{T}}]$$

步骤 2：for $k = 1, 2, \cdots$

（1）计算协方差方根的列矢量用于时间更新，即

$$\boldsymbol{s}_{k-1}^{x,i} = h\left(\sqrt{\boldsymbol{P}_{\boldsymbol{x}_{k-1}}}\right)_i \quad i = 1, 2, \cdots, L_x$$

$$\boldsymbol{s}_{k-1}^{w,i} = h\left(\sqrt{\boldsymbol{R}_{\boldsymbol{w}}}\right)_i \quad i = 1, 2, \cdots, L_v$$

（2）时间更新式，即

$$\begin{aligned}
\hat{\boldsymbol{x}}_{k|k-1} = & \frac{h^2 - L_x - L_w}{h^2} f(\hat{\boldsymbol{x}}_{k-1}, \bar{\boldsymbol{w}}, \boldsymbol{u}_{k-1}) \\
& + \frac{1}{2h^2} \sum_{i=1}^{L_x} [f(\hat{\boldsymbol{x}}_{k-1} + \boldsymbol{s}_{k-1}^{x,i}, \bar{\boldsymbol{w}}, \boldsymbol{u}_{k-1}) + f(\hat{\boldsymbol{x}}_{k-1} - \boldsymbol{s}_{k-1}^{x,i}, \bar{\boldsymbol{w}}, \boldsymbol{u}_{k-1})] \\
& + \frac{1}{2h^2} \sum_{i=1}^{L_w} [f(\hat{\boldsymbol{x}}_{k-1}, \bar{\boldsymbol{w}} + \boldsymbol{s}_{k-1}^{w,i}, \boldsymbol{u}_{k-1}) + f(\hat{\boldsymbol{x}}_{k-1}, \bar{\boldsymbol{w}} - \boldsymbol{s}_{k-1}^{w,i}, \boldsymbol{u}_{k-1})]
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{P}_{k|k-1} = & \frac{1}{4h^2} \sum_{i=1}^{L_x} [f(\hat{\boldsymbol{x}}_{k-1} + \boldsymbol{s}_{k-1}^{x,i}, \bar{\boldsymbol{w}}, \boldsymbol{u}_{k-1}) - f(\hat{\boldsymbol{x}}_{k-1} - \boldsymbol{s}_{k-1}^{x,i}, \bar{\boldsymbol{w}}, \boldsymbol{u}_{k-1})]^2 \\
& + \frac{1}{4h^2} \sum_{i=1}^{L_w} [f(\hat{\boldsymbol{x}}_{k-1}, \bar{\boldsymbol{w}} + \boldsymbol{s}_{k-1}^{w,i}, \boldsymbol{u}_{k-1}) - f(\hat{\boldsymbol{x}}_{k-1}, \bar{\boldsymbol{w}} - \boldsymbol{s}_{k-1}^{w,i}, \boldsymbol{u}_{k-1})]^2 \\
& + \frac{h^2 - 1}{4h^4} \sum_{i=1}^{L_x} [f(\hat{\boldsymbol{x}}_{k-1} + \boldsymbol{s}_{k-1}^{x,i}, \bar{\boldsymbol{w}}, \boldsymbol{u}_{k-1}) + f(\hat{\boldsymbol{x}}_{k-1} - \boldsymbol{s}_{k-1}^{x,i}, \bar{\boldsymbol{w}}, \boldsymbol{u}_{k-1}) - 2f(\hat{\boldsymbol{x}}_{k-1}, \bar{\boldsymbol{w}}, \boldsymbol{u}_{k-1})]^2 \\
& + \frac{h^2 - 1}{4h^4} \sum_{i=1}^{L_w} [f(\hat{\boldsymbol{x}}_{k-1}, \bar{\boldsymbol{w}} + \boldsymbol{s}_{k-1}^{w,i}, \boldsymbol{u}_{k-1}) + f(\hat{\boldsymbol{x}}_{k-1}, \bar{\boldsymbol{w}} - \boldsymbol{s}_{k-1}^{w,i}, \boldsymbol{u}_{k-1}) - 2f(\hat{\boldsymbol{x}}_{k-1}, \bar{\boldsymbol{w}}, \boldsymbol{u}_{k-1})]^2
\end{aligned}$$

（3）计算协方差方根的列矢量用于测量更新，即

$$\boldsymbol{s}_k^{x,i} = h\left(\sqrt{\boldsymbol{P}_{k|k-1}}\right)_i \quad i = 1, 2, \cdots, L_x$$

$$\boldsymbol{s}_k^{v,i} = h\left(\sqrt{\boldsymbol{R}_{\boldsymbol{v}}}\right)_i \quad i = 1, 2, \cdots, L_v$$

（4）测量更新，即

$$
\begin{aligned}
\hat{\boldsymbol{y}}_{k|k-1} &= \frac{h^2 - L_x - L_v}{h^2} h(\hat{\boldsymbol{x}}_{k|k-1}, \bar{\boldsymbol{v}}) \\
&+ \frac{1}{2h^2} \sum_{i=1}^{L_x} [h(\hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{s}_k^{x,i}, \bar{\boldsymbol{v}}) + h(\hat{\boldsymbol{x}}_{k|k-1} - \boldsymbol{s}_k^{x,i}, \bar{\boldsymbol{v}})] \\
&+ \frac{1}{2h^2} \sum_{i=1}^{L_v} [h(\hat{\boldsymbol{x}}_{k|k-1}, \bar{\boldsymbol{v}} + \boldsymbol{s}_k^{x,i}) + h(\hat{\boldsymbol{x}}_{k|k-1}, \bar{\boldsymbol{v}} - \boldsymbol{s}_k^{x,i})] \\
\boldsymbol{P}_{\boldsymbol{y}_k \boldsymbol{y}_k} &= \frac{1}{4h^2} \sum_{i=1}^{L_x} [h(\hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{s}_k^{x,i}, \bar{\boldsymbol{v}}) - f(\hat{\boldsymbol{x}}_{k|k-1} - \boldsymbol{s}_k^{x,i}, \bar{\boldsymbol{v}})]^2 \\
&+ \frac{1}{4h^2} \sum_{i=1}^{L_v} [h(\hat{\boldsymbol{x}}_{k|k-1}, \bar{\boldsymbol{v}} + \boldsymbol{s}_k^{v,i}) - h(\hat{\boldsymbol{x}}_{k|k-1}, \bar{\boldsymbol{v}} - \boldsymbol{s}_k^{v,i})]^2 \\
&+ \frac{h^2 - 1}{4h^4} \sum_{i=1}^{L_x} [h(\hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{s}_k^{x,i}, \bar{\boldsymbol{v}}) + h(\hat{\boldsymbol{x}}_{k|k-1} - \boldsymbol{s}_k^{x,i}, \bar{\boldsymbol{v}}) - 2h(\hat{\boldsymbol{x}}_{k|k-1}, \bar{\boldsymbol{v}})]^2 \\
&+ \frac{h^2 - 1}{4h^4} \sum_{i=1}^{L_v} [h(\hat{\boldsymbol{x}}_{k|k-1}, \bar{\boldsymbol{v}} + \boldsymbol{s}_k^{v,i}) + h(\hat{\boldsymbol{x}}_{k|k-1}, \bar{\boldsymbol{v}} - \boldsymbol{s}_k^{v,i}) - 2h(\hat{\boldsymbol{x}}_{k|k-1}, \bar{\boldsymbol{v}})]^2 \\
\boldsymbol{P}_{\boldsymbol{x}_k \boldsymbol{y}_k} &= \frac{1}{2h^2} \sum_{i=1}^{L_x} \boldsymbol{s}_k^{x,i} [h(\hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{s}_k^{x,i}, \bar{\boldsymbol{v}}) - h(\hat{\boldsymbol{x}}_{k|k-1} - \boldsymbol{s}_k^{x,i}, \bar{\boldsymbol{v}})]^{\mathrm{T}} \\
\boldsymbol{K}_k &= \boldsymbol{P}_{\boldsymbol{x}_k \boldsymbol{y}_k} \boldsymbol{P}_{\boldsymbol{y}_k \boldsymbol{y}_k}^{-1} \\
\hat{\boldsymbol{x}}_k &= \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k (\boldsymbol{y}_k - \hat{\boldsymbol{y}}_{k|k-1}) \\
\boldsymbol{P}_k &= \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k \boldsymbol{P}_{\boldsymbol{y}_k \boldsymbol{y}_k} \boldsymbol{K}_k^{\mathrm{T}}
\end{aligned}
$$

参数说明：上述计算公式中，$h \geqslant 1$ 为差分步长（标量），对于高斯变量 $\boldsymbol{x}$，该参数的最优取值为 $h=\sqrt{3}$；L_x、L_w 和 L_v 分别为状态变量、过程噪声和观测噪声的维数；$\left(\sqrt{\boldsymbol{P}}\right)_i$ 表示矩阵方根 $\sqrt{\boldsymbol{P}}$ 的第 i 列。另外，为表示方便上述诸式中用 $[\boldsymbol{x}]^2$ 代表 $\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}$。

3.3.4 求根中心差分卡尔曼滤波器

各种卡尔曼型滤波算法中，运算开销最大的是协方差矩阵的开根，出于提高运算效率及数值稳定性等目的，文献[8]采用了求根形式的 CDKF 算法（SR-CDKF）。SR-CDKF 算法利用 Cholesky 分解形式直接传递和更新协方差矩阵的方根，与标准的 CDKF 算法相比，SR-CDKF 算法具有相近或略优的估计精度、稍低的运算开销及更好的数值稳定性。

SR-CDKF 的实现主要利用了 QR 分解、Cholesky 因子更新和最小二乘解这三大线性代数工具，其具体的实现细节如下：

1. QR 分解

矩阵 $\boldsymbol{A} \in \mathbb{R}^{L \times N}$ 的 QR 分解可表示为

$$
\boldsymbol{A}^{\mathrm{T}} = \boldsymbol{Q}\boldsymbol{R}
$$

式中：$\boldsymbol{Q} = \mathbb{R}^{N \times N}$ 是正交矩阵；$\mathbb{R}^{N \times L}$ 为上三角阵且 $N \geqslant L$。$\boldsymbol{R}$ 的上三角部分 $\tilde{\boldsymbol{R}}$ 为 Cholesky 因子 $\boldsymbol{P} = \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$ 的转置，即 $\tilde{\boldsymbol{R}} = \boldsymbol{A}^{\mathrm{T}}$，故有 $\tilde{\boldsymbol{R}}^{\mathrm{T}}\tilde{\boldsymbol{R}} = \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$。这里利用缩写形式 $qr\{\cdot\}$ 来标识矩阵的 RQ 分解并返回 $\tilde{\boldsymbol{R}}$。

RQ 分解的运算复杂度为 $O(NL^2)$，对 $\boldsymbol{P} = \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$ 直接利用 Cholesky 因式进行分解的复

杂度为 $O(L^3/6)+O(NL^2)$。

2. Cholesky 因子更新

若 $\boldsymbol{S}$ 为 $\boldsymbol{P}=\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$ 原始的下三角 Cholesky 因子，则 Cholesky 因子的秩−1 更新

$$\boldsymbol{P}'=\boldsymbol{P}\pm\sqrt{\nu}\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}$$

可记为

$$\boldsymbol{S}'=\mathrm{cholupdate}\{\boldsymbol{S},\boldsymbol{u},\pm\nu\}$$

3. 高效最小二乘解

方程 $(\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})\boldsymbol{x}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}$ 的解对应于超定最小二乘问题 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 的解，这可以通过三角 QR 分解求得。

通过上述分析，SR-CDKF 算法可概括如下。

Alg（3.3）：Square-Root Central Difference Kalman Filter

步骤 1：初始化

$$\hat{\boldsymbol{x}}_0=E[\boldsymbol{x}_0]，\ \boldsymbol{S}_{x_0}=chol\{\mathrm{cov}[\boldsymbol{x}_0]\}，\ \boldsymbol{S}_w=\sqrt{\boldsymbol{R}_w}，\ \boldsymbol{S}_v=\sqrt{\boldsymbol{R}_v}$$

步骤 2：for $k=1,2,\cdots$

（1）计算协方差方根的列矢量，即

$$\hat{\boldsymbol{x}}_{k-1}^{a_w}=[\hat{\boldsymbol{x}}_{k-1}\ \bar{\boldsymbol{w}}]，\ \boldsymbol{S}_{k-1}^{a_w}=\begin{bmatrix}\boldsymbol{S}_{x_{k-1}} & 0\\ 0 & \boldsymbol{S}_w\end{bmatrix}$$

$$\boldsymbol{\mathcal{X}}_{k-1}^{a_w}=[\hat{\boldsymbol{x}}_{k-1}^{a_w}\ \ \hat{\boldsymbol{x}}_{k-1}^{a_w}+h\boldsymbol{S}_{k-1}^{a_w}\ \ \hat{\boldsymbol{x}}_{k-1}^{a_w}-h\boldsymbol{S}_{k-1}^{a_w}]$$

（2）时间更新式，即

$$\boldsymbol{\mathcal{X}}_{k|k-1}^{x}=f(\boldsymbol{\mathcal{X}}_{k-1}^{x},\boldsymbol{\mathcal{X}}_{k-1}^{w},\boldsymbol{u}_{k-1})$$

$$\hat{\boldsymbol{x}}_{k|k-1}=\sum_{i=0}^{2L}w_i^{(m)}\boldsymbol{\mathcal{X}}_{i,k|k-1}^{x}$$

$$\boldsymbol{S}_{x_k}^{-}=\mathrm{qr}\left\{\left[\sqrt{w_1^{(c_1)}}\left(\boldsymbol{\mathcal{X}}_{1:L,k|k-1}^{x}-\boldsymbol{\mathcal{X}}_{L+1:2L,k|k-1}^{x}\right)\sqrt{w_2^{(c_2)}}\left(\boldsymbol{\mathcal{X}}_{1:L,k|k-1}^{x}+\boldsymbol{\mathcal{X}}_{L+1:2L,k|k-1}^{x}-\boldsymbol{\mathcal{X}}_{0,k|k-1}^{x}\right)\right]\right\}$$

式中

$$\begin{cases}w_0^{(m)}=\dfrac{(h^2-L_x-L_w)}{h^2},w_i^{(m)}=\dfrac{1}{h^2}\\ w_i^{(c_1)}=\dfrac{1}{4h^2},w_i^{(c_2)}=\dfrac{h^2-1}{4h^4}\end{cases}L=2(L_x+L_w);\ i=1,\ \cdots,\ 2L$$

（3）计算量测样点，即

$$\hat{\boldsymbol{x}}_{k-1}^{a_v}=[\hat{\boldsymbol{x}}_{k|k-1}\ \bar{\boldsymbol{v}}]，\ \boldsymbol{S}_{k|k-1}^{a_v}=\begin{bmatrix}\boldsymbol{S}_{x_k}^{-} & 0\\ 0 & \boldsymbol{S}_v\end{bmatrix}$$

$$\boldsymbol{\mathcal{X}}_{k|k-1}^{a_v}=[\hat{\boldsymbol{x}}_{k|k-1}^{a_v}\ \ \hat{\boldsymbol{x}}_{k|k-1}^{a_v}+h\boldsymbol{S}_{k|k-1}^{a_v}\ \ \hat{\boldsymbol{x}}_{k|k-1}^{a_v}-h\boldsymbol{S}_{k|k-1}^{a_v}]$$

（4）测量更新，即

$$\boldsymbol{\mathcal{Y}}_{k|k-1}=h(\boldsymbol{\mathcal{X}}_{k|k-1}^{x},\boldsymbol{\mathcal{X}}_{k|k-1}^{v})$$

$$\hat{\boldsymbol{y}}_{k|k-1}=\sum_{i=0}^{2L}w_i^{(m)}\boldsymbol{\mathcal{Y}}_{i,k|k-1}$$

$$
\begin{aligned}
&\boldsymbol{S}_{y_k}=\mathrm{qr}\left\{\left[\sqrt{w_1^{(c_1)}}\left(\boldsymbol{\mathcal{Y}}_{1:L,k|k-1}-\boldsymbol{\mathcal{Y}}_{L+1:2L,k|k-1}\right)\ \sqrt{w_2^{(c_2)}}\left(\boldsymbol{\mathcal{Y}}_{1:L,k|k-1}+\boldsymbol{\mathcal{Y}}_{L+1:2L,k|k-1}-2\boldsymbol{\mathcal{Y}}_{0,k|k-1}\right)\right]\right\}\\
&\boldsymbol{P}_{x_k y_k}=\sqrt{w_1^{(c_1)}}\boldsymbol{S}_{x_k}^{-}[\boldsymbol{\mathcal{Y}}_{1:L,k|k-1}-\boldsymbol{\mathcal{Y}}_{L+1:2L,k|k-1}]^{\mathrm{T}}\\
&\boldsymbol{K}_k=(\boldsymbol{P}_{x_k y_k}/\boldsymbol{S}_{y_k}^{\mathrm{T}})/\boldsymbol{S}_{y_k}\\
&\hat{\boldsymbol{x}}_k=\hat{\boldsymbol{x}}_{k|k-1}+\boldsymbol{K}_k(\boldsymbol{y}_k-\hat{\boldsymbol{y}}_{k|k-1})\\
&\boldsymbol{U}=\boldsymbol{K}_k\boldsymbol{S}_{y_k}\\
&\boldsymbol{S}_{x_k}=\mathrm{cholupdate}\{\boldsymbol{S}_{x_k}^{-},\boldsymbol{U},-1\}
\end{aligned}
$$

式中

$$
\begin{cases}
w_0^{(m)}=\dfrac{(h^2-L_x-L_v)}{h^2}, w_i^{(m)}=\dfrac{1}{h^2}\\
w_i^{(c_1)}=\dfrac{1}{4h^2}, w_i^{(c_2)}=\dfrac{h^2-1}{4h^4}
\end{cases}
L=2(L_x+L_v);\ i=1,\cdots,\ 2L
$$

3.4 应用实例

3.4.1 径向速度测量跟踪

为了对 EKF 算法及 CDKF 算法进行测试，考虑三维空间中的跟踪问题，目标运动模型采用直角坐标系中的近匀速模型，设k时刻的目标状态为$\boldsymbol{x}_k=[x\ \dot{x}\ y\ \dot{y}\ z\ \dot{z}]^{\mathrm{T}}(k)$，则其状态演化过程可表示为

$$
\boldsymbol{x}_k=\boldsymbol{F}_{CV}\boldsymbol{x}_{k-1}+\boldsymbol{w}_k=\begin{bmatrix}\boldsymbol{F}_1 & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{F}_1 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{F}_1\end{bmatrix}\boldsymbol{x}_{k-1}+\boldsymbol{w}_k \tag{3-74}
$$

$$
\boldsymbol{Q}_k=\mathrm{cov}[\boldsymbol{w}_k]=q^2\cdot\begin{bmatrix}\boldsymbol{Q}_1 & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{Q}_1 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{Q}_1\end{bmatrix} \tag{3-75}
$$

式中

$$
\boldsymbol{F}_1=\begin{bmatrix}1 & T\\ 0 & 1\end{bmatrix},\ \boldsymbol{Q}_1=\begin{bmatrix}T^3/3 & T^2/2\\ T^2/2 & T\end{bmatrix}
$$

T为采样间隔；$\boldsymbol{w}_k$为零均值高斯过程噪声。

采用主动雷达传感器，观测量包括目标距离r_k、方位φ_k、俯仰θ_k及径向速度$\dot{r}_k$（或称距离变化率），观测点位于坐标原点。观测方程可表示为

$$
\boldsymbol{y}_k=h(\boldsymbol{x}_k)+\boldsymbol{v}_k=\begin{bmatrix}\sqrt{x_k^2+y_k^2+z_k^2}\\ \arctan(y_k/x_k)\\ \arctan(z_k/\sqrt{x_k^2+y_k^2})\\ \dfrac{x_k\dot{x}_k+y_k\dot{y}_k+z_k\dot{z}_k}{\sqrt{x_k^2+y_k^2+z_k^2}}\end{bmatrix}+\begin{bmatrix}v_{r,k}\\ v_{\varphi,k}\\ v_{\theta,k}\\ v_{\dot{r},k}\end{bmatrix} \tag{3-76}
$$

式中：$\boldsymbol{v}_k$ 为零均值高斯噪声，且 $\text{cov}[\boldsymbol{v}_k]=\boldsymbol{R}_k$。

假定目标初始位置为(30,45,10)km，初始速度为(−160, −120,0)m/s，目标状态的初始估计误差协方差矩阵为 $\boldsymbol{P}_0=\text{diag}([250^2,100^2,250^2,100^2,250^2,50^2])$，其他参数见表 3.3。采用随机初始化，产生 500 次实验数据，利用 Alg（3.1）和 Alg（3.2）中的 EKF 和 CDKF 进行跟踪（两种算法的初始条件完全相同），得到的结果如图 3.2~图 3.5 所示。

表 3.3　滤波器参数设置

q	T	σ_r	σ_φ	σ_θ	$\sigma_{\dot{r}}$	h
1m/s^2	2s	50m	0.3°	0.3°	2m/s	$\sqrt{3}$

图 3.2 为单次跟踪的航迹估计示例，图 3.3 为 500 次仿真得到的归一化估计方差（NEES），图 3.4 和图 3.5 分别为位置和速度估计均方根误差（RMSE）。由图 3.2～图 3.5 中的结果可以看出，在本实例参数设置条件下，EKF 和 CDKF 性能几乎完全相当，且目标位置和速度跟踪都有明显的收敛过程，这说明在该条件下两种算法的近似精度是等效的。较之位置估计，速度估计的收敛速度更快，这是因为增加了径向速度观测量信息。另外，从图 3.3 中的结果看，跟踪收敛后两种算法的 NEES 均在 6 附近以微小的幅度起伏（理论值为 6），说明跟踪算法具有良好的估计一致性。

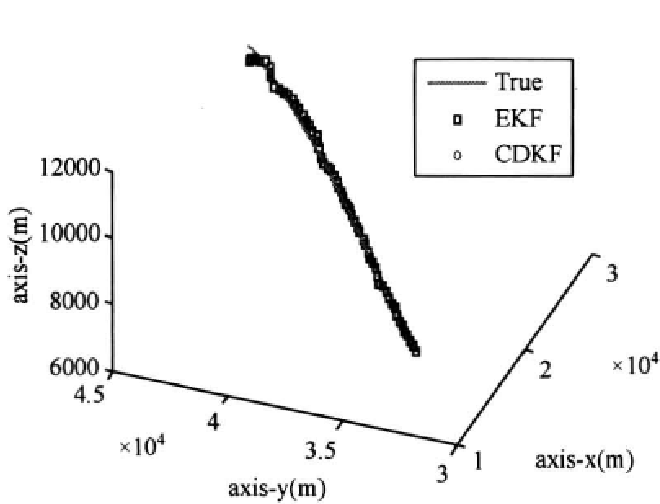


图 3.2　单次跟踪示例

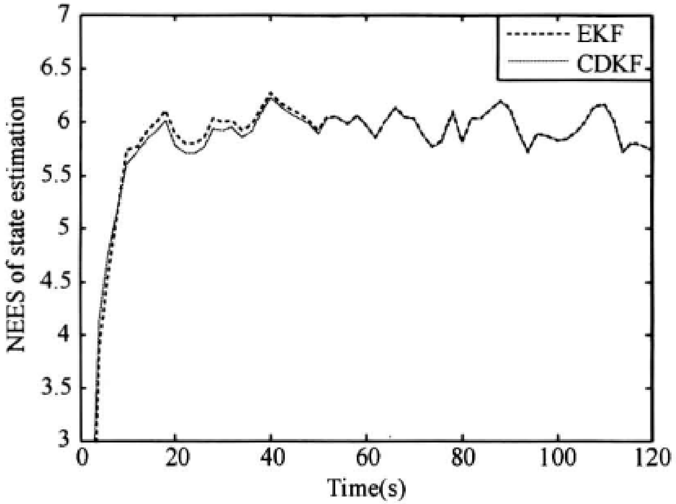


图 3.3　归一化估计方差

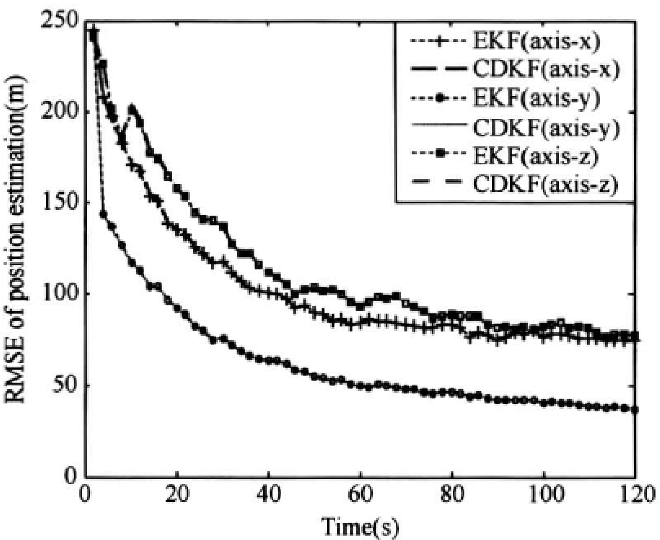


图 3.4　位置均方根误差

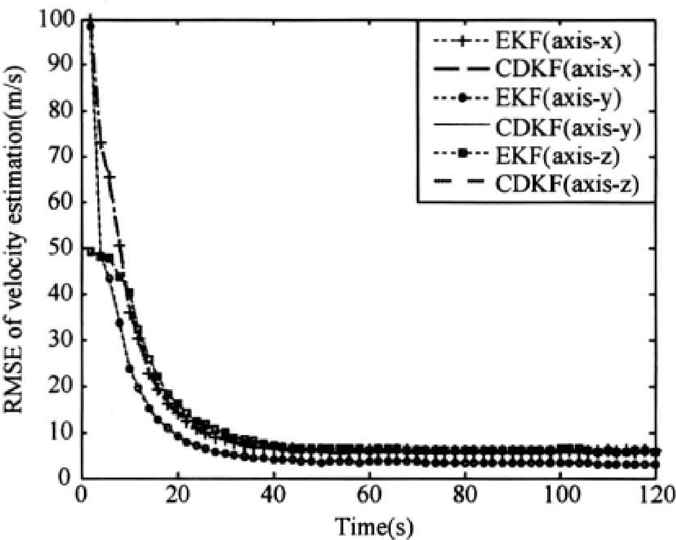


图 3.5　速度均方根误差

3.4.2 椭圆扩展目标跟踪

考虑直角坐标系中二维平面内的目标运动，若目标状态（包括位置和速度）为 $\boldsymbol{x}_k = [x\ \dot{x}\ y\ \dot{y}]^{\mathrm{T}}(k)$，则当目标作近匀速（CV）运动时，其状态方程可表示为

$$\boldsymbol{x}_k = \boldsymbol{F}_{CV}\boldsymbol{x}_{k-1} + \boldsymbol{w}_k = \begin{bmatrix} \boldsymbol{F}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{F}_1 \end{bmatrix} \boldsymbol{x}_{k-1} + \boldsymbol{w}_k \tag{3-77}$$

$$\boldsymbol{Q}_k = \mathrm{cov}[\boldsymbol{w}_k] = q^2 \cdot \begin{bmatrix} \boldsymbol{Q}_1 & 0 \\ 0 & \boldsymbol{Q}_1 \end{bmatrix} \tag{3-78}$$

式中

$$\boldsymbol{F}_1 = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix},\quad \boldsymbol{Q}_1 = \begin{bmatrix} T^3/3 & T^2/2 \\ T^2/2 & T \end{bmatrix}$$

T 为离散采样间隔。

采用雷达观测得到目标距离 r_k、方位 φ_k 及径向长度 l_k 信息，观测方程为

$$\boldsymbol{y}_k = h(\boldsymbol{x}_k) + \boldsymbol{v}_k = \begin{bmatrix} \sqrt{x_k^2 + y_k^2} \\ \arctan(y_k / x_k) \\ \ell\sqrt{\cos^2\phi_k + (b/a)^2 \sin^2\phi_k} \end{bmatrix} + \begin{bmatrix} v_{r,k} \\ v_{\varphi,k} \\ v_{l,k} \end{bmatrix} \tag{3-79}$$

式中：ℓ 为目标主轴向长度；b/a 为椭圆目标的短轴与长轴长度比；$\boldsymbol{v}_k$ 为零均值高斯噪声且 $\mathrm{cov}[\boldsymbol{v}_k] = \boldsymbol{R}_k$，$\phi_k$ 为方向角（雷达视线与目标速度方向的夹角）且

$$\phi_k = \arccos \frac{x_k \dot{x}_k + y_k \dot{y}_k}{\sqrt{x_k^2 + y_k^2}\sqrt{\dot{x}_k^2 + \dot{y}_k^2}} \tag{3-80}$$

进一步假定 b/a 已知，而 ℓ 为未知量，为此需要将 ℓ 与 $\boldsymbol{x}_k$ 组合成扩展状态量 $\boldsymbol{x}_k^a = [\boldsymbol{x}_k^{\mathrm{T}}\ l_k]^{\mathrm{T}}$，此时跟踪算法中的状态模型为

$$\boldsymbol{x}_k^a = \boldsymbol{F}\boldsymbol{x}_{k-1}^a + \boldsymbol{w}_k^a = \begin{bmatrix} \boldsymbol{F}_1 & \boldsymbol{0} & 0 \\ \boldsymbol{0} & \boldsymbol{F}_1 & 0 \\ \boldsymbol{0} & \boldsymbol{0} & 1 \end{bmatrix} \boldsymbol{x}_{k-1}^a + \boldsymbol{w}_k^a \tag{3-81}$$

式中

$$\boldsymbol{w}_k^a = [\boldsymbol{w}_k^{\mathrm{T}}\ w_{l,k}]^{\mathrm{T}},\quad \mathrm{cov}[\boldsymbol{w}_k^a] = \boldsymbol{Q}_k^a = \mathrm{diag}([\boldsymbol{Q}_k\ q_l^2 T^2])$$

对于式（3-79）和式（3-81）组成的系统模型，由于 $l(\phi_k) = \ell\sqrt{\cos^2\phi_k + (b/a)^2\sin^2\phi_k}$ 是系统状态的隐函数，在使用 EKF 算法进行跟踪时，求 Jacobian 矩阵是较为复杂的，此时可采用偏导的链规则来计算，即

$$\frac{\partial l(\phi)}{\partial \tau} = \frac{\partial l(\phi)}{\partial \phi}\frac{\partial \phi}{\partial \tau},\quad \tau = x, \dot{x}, y, \dot{y} \tag{3-82}$$

实验中，假定目标初始状态为 $\boldsymbol{x}_0^a = [20\mathrm{km}\ \ 10\sqrt{3}\mathrm{m/s}\ \ 15\mathrm{km}\ \ 10\mathrm{m/s}\ \ 50\mathrm{m}]^{\mathrm{T}}$，目标状态的初始估计误差协方差矩阵为 $\boldsymbol{P}_0 = \mathrm{diag}([100^2, 10^2, 100^2, 10^2, 10^2])$，其他参数设置见表 3.4。

表 3.4 仿真参数设置

q	q_ℓ	T	σ_r	σ_φ	σ_l	ℓ	b/a	h
0.1m/s^2	0.001m/s	2s	5m	$0.3°$	2m	50m	0.3	$\sqrt{3}$

根据以上条件，采用随机初始化进行 500 次独立实验，并利用 EKF 和 CDKF 进行跟踪，所得结果如图 3.6～图 3.10 所示。

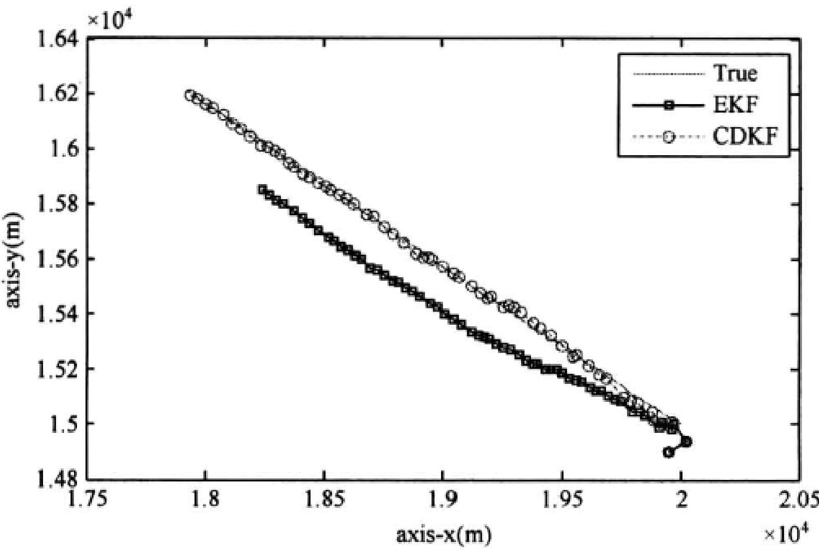


图 3.6 单次跟踪示例

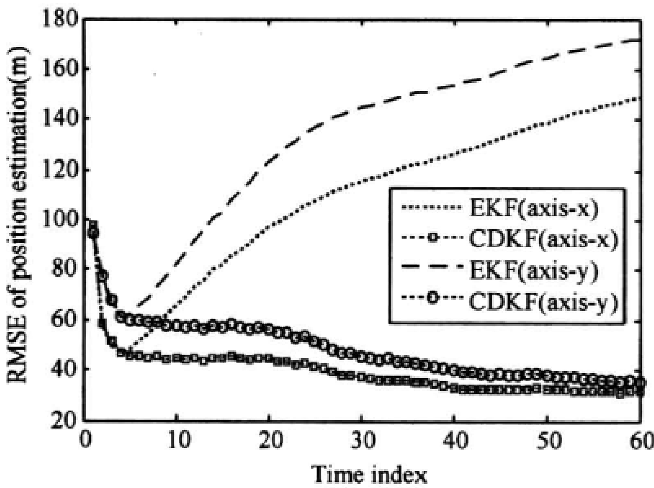


图 3.7 位置估计均方根误差

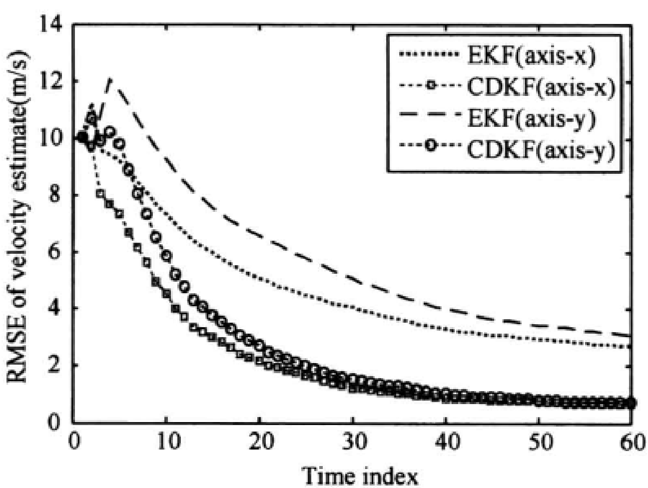


图 3.8 速度估计均方根误差

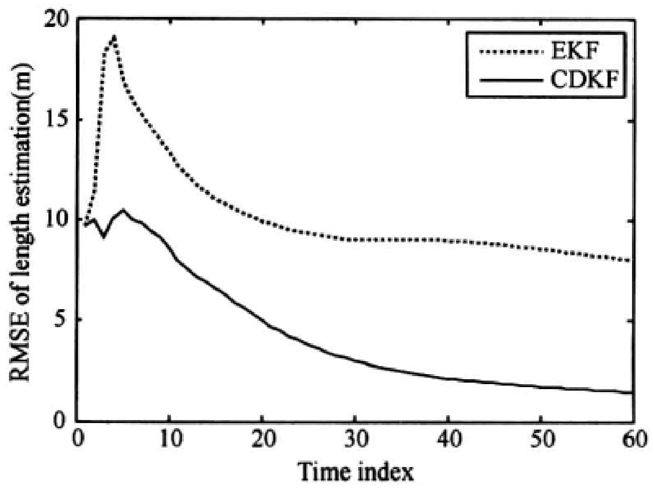


图 3.9 长度估计均方根误差

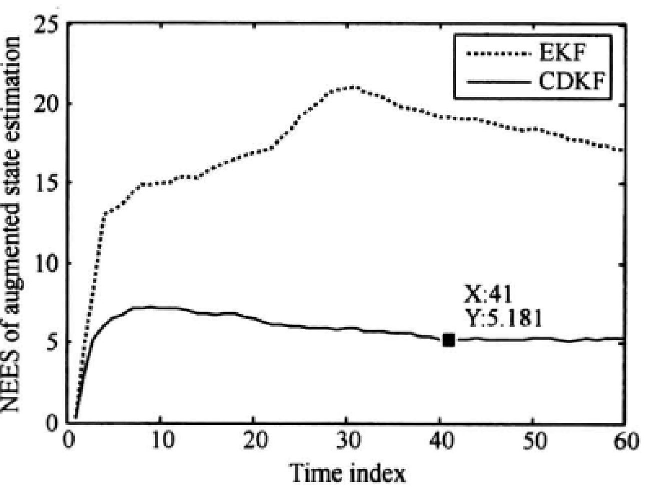


图 3.10 归一化估计方差

由图 3.6～图 3.10 可以看出，与前一个示例中的结果明显不同，本实例中 EKF 和 CDKF 的性能差异较大，这一点无论在单次跟踪还是多次统计结果中都有具体体现。究其原因，主要是观测方程中的目标长度观测函数具有较强的非线性，在这种条件下基于一阶泰勒展开的 EKF 算法近似误差较大；同时，目标运动状态与长度信息在求取 Jacobian 矩阵时存在强耦合，进一步引入了误差源。相比之下，具有二阶插值精度的 CDKF 算法近似误差要小得多且不存在偏导数计算环节，因此估计性能要比 EKF 好得多。从图 3.10 中的结果看，在 CDKF 滤波算法收敛后其 NEES 接近理论值 5，说明在本实例条件下仍有较好的估计一致性。

另外，为了比较算法的复杂度，表 3.5 给出了两种算法在 Matlab7.12 环境、2.8GHz 主频和 2.75GB 内存条件下（后面章节中的实验均采用这一环境）500 次仿真的运行时间统计。由此可以看出，两种算法都具有较好的实时性，但相比之下，CDKF 的算法复杂度稍大（本例子中约为 EKF 的 4 倍）。

表 3.5　两种算法运行时间比较

算法 / 执行时间	EKF	CDKF
Mean/（s/run）	0.0044	0.0174
STD/（s/run）	4.84e-4	7.05e-4

3.5 本章小结

EKF 的提出是为了简化待处理的非线性系统以适应线性卡尔曼滤波器的要求，即通过在当前状态线性展开来近似非线性函数。理论和实践均表明，在系统非线性程度不太高且初始误差不大的情况下，EKF 可得得到良好的估计性能（即在满足局部线性化条件下其性能是可靠的）。然而，当应对强非线性系统时，EKF 常常难以调节，有时甚至发散；同时，由于涉及偏导数计算，这种算法只适用于具有显式表达式且偏导数存在的系统方程。

作为非线性函数的一种有效近似手段，Sterling 插值方法具有近似精度高、免偏导数计算等优点，以此为基础的 CDKF 算法在处理非线性程度较高时的滤波问题中较 EKF 算法精度具有明显的优势，但其算法复杂度要比 EKF 高。因此，实际应用中需要根据具体的问题选择合适的方法；另外，从工程实现的角度看，为了提高系统的稳定性，采用求根形式的实现方法[9]更为合适。

需要指出的是，对于 EKF，除了前文提及的几种常见衍生算法以外，还存在衰减记忆 EKF[10]、强跟踪 EKF[11]等几种重要的衍生形式。此外，本章在介绍各种算法时，均以高斯白噪声为假设前提，关于色噪声条件下的卡尔曼滤波算法，可参阅文献[12]，在此不再细述。

第 4 章　确定性采样近似高斯滤波

从第 3 章的分析中可以看出，EKF 及其衍生算法尽管在线性化过程中选择的展开点不同，但都无一例外地要计算 Jacobian 矩阵。用泰勒展开来近似非线性函数的方法往往在对状态后验分布的估计中引入较大的误差，这一问题在模型的非线性程度较高时更为突出，即此时泰勒展开的高阶项影响变得更为显著。由此带来的直接后果是近似精度低、协方差易出现病态，严重时致使滤波器发散。为避免以上问题，文献[13]中提出一种新的基于确定性样点计算的滤波算法，由于该算法建立在不敏变换（Unscented Transform，UT）基础之上，称为不敏卡尔曼滤波（Unscented Kalman Filter，UKF）[14,15]。

与 EKF 不同的是，UKF 并没有近似非线性状态和观测模型，而直接利用真实的系统模型。与 EKF 相同的是，UKF 也将状态近似为高斯随机变量（GRV），但此时这些 GRV 用一组有效的采样点集来表示，这些样本点能捕获到 GRV 的均值和协方差。当状态量通过实际的非线性系统之后，后验均值和协方差可以精确到三阶（泰勒展开），且对任何非线性系统都有这一近似精度。同时，由于不需要推导和计算复杂的 Jacobian/Hessian 矩阵，基于 UT 的 UKF 算法更便于实现。UT 也可视为随机线性化的离散实现[16]，与 EKF 中的局部线性化不同，随机线性化利用了系统在状态空间中的多点信息，这也正好说明了 UKF 精度高于 EKF 的原因。

本章首先介绍不敏变换的基本原理，并对随机变量在非线性系统传递过程中的误差进行了分析，在此基础上，结合卡尔曼滤波框架得到了 UKF 算法。针对具体的应用条件和特定的系统模型，本章还研究了 UKF 的几种重要的扩展形式，如求根 UKF、迭代 UKF、简化 UKF 及联合估计 UKF 等，最后给出了几个典型的应用实例。

4.1　不敏变换与 UKF 滤波

4.1.1　不敏变换 UT

UT 是一种计算随机变量经非线性传递后的统计特性的方法。考虑一个 L 维随机变量 $\boldsymbol{x}$ 经过非线性变换后得到 $\boldsymbol{y}=g(\boldsymbol{x})$，并假设 $\boldsymbol{x}$ 的均值和方差分别为 $\bar{\boldsymbol{x}}$ 和 $\boldsymbol{P}_{\boldsymbol{x}}$。为了计算变量 $\boldsymbol{y}$ 的统计特性，需要选取 $2L+1$ 个样点 χ_i。这些样点的取值规则为

$$\begin{cases}\chi_i=\bar{\boldsymbol{x}} \\ \chi_i=\bar{\boldsymbol{x}}+(\sqrt{(L+\kappa)\boldsymbol{P}_{\boldsymbol{x}}})_i & (i=1,\ \cdots,\ L) \\ \chi_i=\bar{\boldsymbol{x}}-(\sqrt{(L+\kappa)\boldsymbol{P}_{\boldsymbol{x}}})_{i-L} & (i=L+1,\ \cdots,\ 2L)\end{cases} \tag{4-1}$$

$$\begin{cases} w_0^{(m)} = \kappa/(L+\kappa) \\ w_0^{(c)} = \kappa/(L+\kappa) \\ w_i^{(m)} = w_i^{(c)} = 1/(2L+2\kappa) \end{cases} \quad (i=1,\ \cdots,\ 2L) \tag{4-2}$$

式中：$\left(\sqrt{\boldsymbol{P}_x}\right)^{\mathrm{T}}\left(\sqrt{\boldsymbol{P}_x}\right)=\boldsymbol{P}_x$；$\kappa$ 用于确定样点与均值 $\bar{\boldsymbol{x}}$ 之间的距离；$\left(\sqrt{\boldsymbol{P}_x}\right)_i$ 表示矩阵方根的第 i 列。将这些样点通过非线性变换，即

$$\mathcal{Y}_i = g(\chi_i) \quad (i=0,\ 1,\ \cdots,\ 2L) \tag{4-3}$$

变量 $\boldsymbol{y}$ 的均值和方差可计算为

$$\bar{\boldsymbol{y}} = \sum_{i=0}^{2L} w_i^{(m)} \mathcal{Y}_i \tag{4-4}$$

$$\boldsymbol{P}_y = \sum_{i}^{2L} w_i^{(c)} \{\mathcal{Y}_i - \bar{\boldsymbol{y}}\}\{\mathcal{Y}_i - \bar{\boldsymbol{y}}\}^{\mathrm{T}} \tag{4-5}$$

4.1.2 UT 近似误差分析

为分析 UT 的近似精度，首先选择一组样点

$$\chi_i = \bar{\boldsymbol{x}} \pm (\sqrt{(L+\kappa)\boldsymbol{P}_x})_i = \bar{\boldsymbol{x}} \pm \sqrt{(L+\kappa)}\boldsymbol{e}_i = \bar{\boldsymbol{x}} \pm \boldsymbol{\sigma}_i \tag{4-6}$$

式中：$\boldsymbol{e}_i$ 为方根矩阵 $\boldsymbol{P}_x$ 的第 i 列，即有 $\sum_{i=1}^{L}\boldsymbol{e}_i\boldsymbol{e}_i^{\mathrm{T}} = \boldsymbol{P}_x$，$\boldsymbol{\sigma}_i = \sqrt{(L+\kappa)}\boldsymbol{e}_i$。

将式（4-6）代入到多维泰勒展开式（3-2），有

$$\mathcal{Y}_i = g(\chi_i) = g(\bar{\boldsymbol{x}}) + D_{\sigma}g + \frac{1}{2!}D_{\sigma}^2 g + \frac{1}{3!}D_{\sigma}^3 g + \frac{1}{4!}D_{\sigma}^4 g + \cdots \tag{4-7}$$

$$\begin{aligned} \bar{\boldsymbol{y}} &= E[g(\bar{\boldsymbol{x}}+\boldsymbol{e})] \\ &= \frac{\kappa}{L+\kappa}g(\bar{\boldsymbol{x}}) + \frac{1}{2(L+\kappa)} \\ &\quad \sum_{i=1}^{2L}\left(g(\bar{\boldsymbol{x}}) + D_{\sigma_i}g + \frac{1}{2!}D_{\sigma_i}^2 g + \frac{1}{3!}D_{\sigma_i}^3 g + \frac{1}{4!}D_{\sigma_i}^4 g + \cdots\right) \\ &= g(\bar{\boldsymbol{x}}) + \frac{1}{2(L+\kappa)}\sum_{i=1}^{2L}\left(\frac{1}{2!}D_{\sigma_i}^2 g + \frac{1}{4!}D_{\sigma_i}^4 g + \frac{1}{6!}D_{\sigma_i}^6 g + \cdots\right) \end{aligned} \tag{4-8}$$

又

$$\begin{aligned} \sum_{i=1}^{2L} D_{\sigma_i}^2 g &= \sum_{i=1}^{2L}\left[D_{\sigma_i}(D_{\sigma_i}g)^{\mathrm{T}}\right] = \sum_{i=1}^{2L}\left[(\nabla^{\mathrm{T}}\sqrt{(L+\kappa)}\boldsymbol{e}_i\boldsymbol{e}_i^{\mathrm{T}}\sqrt{(L+\kappa)}\nabla)g(\boldsymbol{x})\big|_{x=\bar{x}}\right] \\ &= (L+\kappa)\sum_{i=1}^{2L}\left[(\nabla^{\mathrm{T}}\boldsymbol{e}_i\boldsymbol{e}_i^{\mathrm{T}}\nabla)g(\boldsymbol{x})\big|_{x=\bar{x}}\right] = 2(L+\kappa)(\nabla^{\mathrm{T}}\boldsymbol{P}_x\nabla)g(\boldsymbol{x})\big|_{x=\bar{x}} \end{aligned} \tag{4-9}$$

于是

$$\bar{\boldsymbol{y}}^{\mathrm{UT}} = g(\bar{\boldsymbol{x}}) + \frac{1}{2}(\nabla^{\mathrm{T}}\boldsymbol{P}_x\nabla)g(\boldsymbol{x})\big|_{x=\bar{x}} + \frac{1}{2(L+\kappa)}\sum_{i=1}^{2L}\left(\frac{1}{4!}D_{\sigma_i}^4 g + \frac{1}{6!}D_{\sigma_i}^6 g + \cdots\right) \tag{4-10}$$

同理

$$
\begin{aligned}
\boldsymbol{P}_y^{\mathrm{UT}} &= E[(\boldsymbol{y}-\overline{\boldsymbol{y}})(\boldsymbol{y}-\overline{\boldsymbol{y}})^{\mathrm{T}}] \\
&= \boldsymbol{G}_{\overline{x}}\boldsymbol{P}_x\boldsymbol{G}_{\overline{x}}^{\mathrm{T}} - \frac{1}{4}\left(\boldsymbol{G}_{\overline{x}}\boldsymbol{P}_x\boldsymbol{G}_{\overline{x}}^{\mathrm{T}}\right)\left(\boldsymbol{G}_{\overline{x}}\boldsymbol{P}_x\boldsymbol{G}_{\overline{x}}^{\mathrm{T}}\right)^{\mathrm{T}} \\
&+ \frac{1}{2(L+\kappa)}\sum_{l=1}^{2L}\underbrace{\sum_{i=1}^{\infty}\sum_{j=1}^{\infty}\frac{1}{i!\,j!}D_{\sigma_l}^{i}g\left(D_{\sigma_l}^{j}g\right)^{\mathrm{T}}}_{\forall i,j:\ ij>1} \\
&- \underbrace{\sum_{i=1}^{\infty}\sum_{j=1}^{\infty}\frac{\sum_{l=1}^{2L}\sum_{m=1}^{2L}D_{\sigma_l}^{2i}g\left(D_{\sigma_m}^{2j}g\right)^{\mathrm{T}}}{4(2i)!(2j)!(L+\kappa)^2}}_{\forall i,j:\ ij>1}
\end{aligned}
\tag{4-11}
$$

对比式（4-10）和式（3-7）不难看出，当 $\boldsymbol{x}$ 为高斯变量时，经 UT 求得的后验均值可以达到三阶精度，仅在四阶以上展开项存在误差；同理，对比式（4-11）和式（3-9）可以看出，后验协方差也可精确到二阶。相比之下，线性化所得的结果无论在均值还是协方差都只有一阶精度。由推导过程同样可以看出，UT 所能达到的精度不依赖于方程的具体形式。这就说明，对于任意的非线性系统，都能达到同样的近似效果。

为提高变换精度，减小高阶矩近似误差，可通过引入一尺度因子 α（$0<\alpha\leqslant 1$），使新的变换样点满足 $\tilde{\chi}_i=\chi_0+\alpha(\chi_i-\chi_0)$，并计算相应的权值来实现。这种方法称为变尺度 UT（Scaled UT），其具体实现过程可参阅文献[17]、文献[18]，这里不再赘述。

作为比较，图 4.1 示出了几种不同变换方法的近似误差示例。由图中的结果可以看到，GRV 经非线性系统传递后，由一阶线性化方法得到的后验均值和协方差估计与真值相差甚远；相比之下，由 UT 得到的估计与真值则较为接近，这说明 UT 的近似精度要明显高于线性化方法。

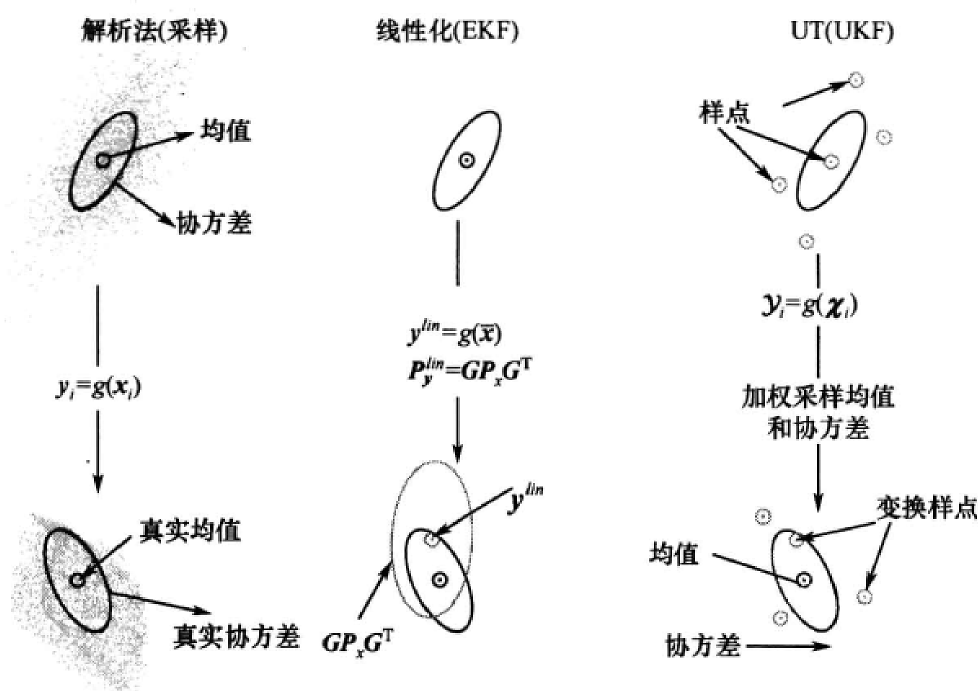


图 4.1　非线性变换比较

下面仍已以式（3-31）的非线性方程为例，分析 UT 所能达到的精度。此时 $L=1$，由式（4-3）可得到 3 个样点，即

$$\chi=\{x, x+\sigma_x, x-\sigma_x\} \tag{4-12}$$

经式（4-3）传递后得

$$\mathcal{Y}=\{a\bar{x}^2+b, a(\bar{x}+\sqrt{\kappa+1}\sigma_x)^2+b, a(\bar{x}-\sqrt{\kappa+1}\sigma_x)^2+b\} \tag{4-13}$$

再由式（4-4）和式（4-5），得

$$\begin{aligned}\bar{y}^{\mathrm{UT}}&=\frac{\kappa}{1+\kappa}(a\bar{x}^2+b)+\frac{1}{2(1+\kappa)}\{[a(\bar{x}+\sqrt{\kappa+1}\sigma_x)^2+b]+\\&\quad[a(\bar{x}-\sqrt{\kappa+1}\sigma_x)^2+b]\}\\&=\frac{\kappa}{1+\kappa}(a\bar{x}^2+b)+\frac{1}{2(1+\kappa)}[2(a\bar{x}^2+b)+2a(\kappa+1)\sigma_x^2]\\&=a(\bar{x}^2+\sigma_x^2)+b\end{aligned} \tag{4-14}$$

$$\begin{aligned}P_{\mathrm{y}}^{\mathrm{UT}}&=\frac{\kappa}{1+\kappa}(a\sigma_x^2)^2+\frac{a^2}{2(1+\kappa)}\{[2\bar{x}\sqrt{\kappa+1}\sigma_x+\kappa\sigma_x^2]^2+\\&\quad[-2\bar{x}\sqrt{\kappa+1}\sigma_x+\kappa\sigma_x^2]^2\}\\&=\frac{\kappa}{1+\kappa}a^2\sigma_x^4+\frac{2a^2}{2(1+\kappa)}[4\bar{x}^2(\kappa+1)\sigma_x^2+\kappa^2\sigma_x^4]\\&=\kappa a^2\sigma_x^4+4a^2\bar{x}^2\sigma_x^2\end{aligned} \tag{4-15}$$

由此可见，经 UT 得到的均值与真实值完全一致；而方差的近似精度与因子κ的选择有关，在本例中当$\kappa=2$时，所得的协方差也是精确的。

4.1.3 UKF 算法

考虑如下一般形式的非线性系统方程：

$$\boldsymbol{x}_k=f(\boldsymbol{x}_{k-1}, \boldsymbol{w}_k) \tag{4-16}$$

$$\boldsymbol{y}_k=h(\boldsymbol{x}_k, \boldsymbol{v}_k) \tag{4-17}$$

式中：$f(\cdot)$和$h(\cdot)$为非线性函数；$\boldsymbol{w}_k\in\mathbb{R}^{L_w}$和$\boldsymbol{v}_k\in\mathbb{R}^{L_v}$分别为过程噪声和观测噪声，且相应的协方差矩阵为$\boldsymbol{Q}_k$和$\boldsymbol{R}_k$。

将 4.1.1 节介绍的 UT 融合到 KF 中，便可得到如下 UKF 算法。

Alg（4.1）：Unscented Kalman Filtering

步骤 1：初始化

$$\hat{\boldsymbol{x}}_0=E[\boldsymbol{x}_0] \tag{4-18}$$

$$\boldsymbol{P}_0=E\left[(\boldsymbol{x}_0-\hat{\boldsymbol{x}}_0)(\boldsymbol{x}_0-\hat{\boldsymbol{x}}_0)^{\mathrm{T}}\right] \tag{4-19}$$

$$\hat{\boldsymbol{x}}_0^a=E\left[\boldsymbol{x}^a\right]=\begin{bmatrix}\hat{\boldsymbol{x}}_0^{\mathrm{T}} & \mathbf{0} & \mathbf{0}\end{bmatrix}^{\mathrm{T}} \tag{4-20}$$

$$\boldsymbol{P}_0^a=E\left[(\boldsymbol{x}_0^a-\hat{\boldsymbol{x}}_0^a)(\boldsymbol{x}_0^a-\hat{\boldsymbol{x}}_0^a)^{\mathrm{T}}\right]=\begin{bmatrix}\boldsymbol{P}_0 & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{Q}_0 & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \boldsymbol{R}_0\end{bmatrix} \tag{4-21}$$

步骤 2：样点计算

$$\boldsymbol{\chi}_{k-1}^{a}=\left[\hat{\boldsymbol{x}}_{k-1}^{a} \quad \hat{\boldsymbol{x}}_{k-1}^{a} \pm \sqrt{\left(L^{a}+\kappa\right) \boldsymbol{P}_{k-1}^{a}}\right] \tag{4-22}$$

步骤 3：时间更新

$$\boldsymbol{\chi}_{k|k-1}^{x}=f\left(\boldsymbol{\chi}_{k-1}^{x}, \boldsymbol{\chi}_{k-1}^{w}\right) \tag{4-23}$$

$$\hat{\boldsymbol{x}}_{k|k-1}=\sum_{i=0}^{2 L_{a}} w_{i}^{(m)} \boldsymbol{\chi}_{i, k|k-1}^{x} \tag{4-24}$$

$$\boldsymbol{P}_{k|k-1}=\sum_{i=0}^{2 L_{a}} w_{i}^{(c)}\left[\boldsymbol{\chi}_{i, k|k-1}^{x}-\hat{\boldsymbol{x}}_{k|k-1}\right]\left[\boldsymbol{\chi}_{i, k|k-1}^{x}-\hat{\boldsymbol{x}}_{k|k-1}\right]^{\mathrm{T}} \tag{4-25}$$

$$\boldsymbol{\mathcal{Y}}_{k|k-1}=h\left(\boldsymbol{\chi}_{k|k-1}^{x}, \boldsymbol{\chi}_{k-1}^{v}\right) \tag{4-26}$$

$$\hat{\boldsymbol{y}}_{k|k-1}=\sum_{i=0}^{2 L_{a}} w_{i}^{(m)} \boldsymbol{\mathcal{Y}}_{i, k|k-1} \tag{4-27}$$

步骤 4：测量更新

$$\boldsymbol{P}_{y_k y_k}=\sum_{i=0}^{2 L_{a}} w_{i}^{(c)}[\boldsymbol{\mathcal{Y}}_{k, k|k-1}-\hat{\boldsymbol{y}}_{k|k-1}][\boldsymbol{\mathcal{Y}}_{i, k|k-1}-\hat{\boldsymbol{y}}_{k|k-1}]^{\mathrm{T}} \tag{4-28}$$

$$\boldsymbol{P}_{x_k y_k}=\sum_{i=0}^{2 L_{a}} w_{i}^{(c)}[\boldsymbol{\chi}_{i, k|k-1}^{x}-\hat{\boldsymbol{x}}_{k|k-1}][\boldsymbol{\mathcal{Y}}_{i, k|k-1}-\hat{\boldsymbol{y}}_{k|k-1}]^{\mathrm{T}} \tag{4-29}$$

$$\boldsymbol{K}_{k}=\boldsymbol{P}_{x_k y_k} \boldsymbol{P}_{y_k y_k}^{-1} \tag{4-30}$$

$$\hat{\boldsymbol{x}}_{k}=\hat{\boldsymbol{x}}_{k|k-1}+\boldsymbol{K}_{k}(\boldsymbol{y}_{k}-\hat{\boldsymbol{y}}_{k|k-1}) \tag{4-31}$$

$$\boldsymbol{P}_{k}=\boldsymbol{P}_{k|k-1}-\boldsymbol{K}_{k} \boldsymbol{P}_{y_k y_k} \boldsymbol{K}_{k}^{\mathrm{T}} \tag{4-32}$$

式中

$$\boldsymbol{\chi}^{a}=[(\boldsymbol{\chi}^{x})^{\mathrm{T}} \ (\boldsymbol{\chi}^{w})^{\mathrm{T}} \ (\boldsymbol{\chi}^{v})^{\mathrm{T}}]^{\mathrm{T}}, \quad \boldsymbol{x}^{a}=[\boldsymbol{x}^{\mathrm{T}} \ \boldsymbol{w}^{\mathrm{T}} \ \boldsymbol{v}^{\mathrm{T}}]^{\mathrm{T}}, \quad L_{a}=L_{x}+L_{w}+L_{v}$$

4.2 求根 UKF 算法

受有限字长等因素的影响，工程应用中为了避免因舍入误差导致的数值不稳定问题，常常采用求根形式的 UKF（SR-UKF）[19]，来避免矩阵求逆操作，提高滤波算法的数值稳定性。SR-UKF 的算法流程可概括如下。

Alg（4.2）： Square Root Unscented Kalman Filtering

步骤 1：初始化

$$\hat{\boldsymbol{x}}_{0}=E\left[\boldsymbol{x}_{0}\right]$$

$$\boldsymbol{P}_{0}=E\left[\left(\boldsymbol{x}_{0}-\hat{\boldsymbol{x}}_{0}\right)\left(\boldsymbol{x}_{0}-\hat{\boldsymbol{x}}_{0}\right)^{\mathrm{T}}\right]$$

$$\hat{\boldsymbol{x}}_{0}^{a}=E\left[\boldsymbol{x}^{a}\right]=\left[\hat{\boldsymbol{x}}_{0}^{\mathrm{T}} \quad \mathbf{0} \quad \mathbf{0}\right]^{\mathrm{T}}$$

$$\boldsymbol{P}_{0}^{a}=E\left[\left(\boldsymbol{x}_{0}^{a}-\hat{\boldsymbol{x}}_{0}^{a}\right)\left(\boldsymbol{x}_{0}^{a}-\hat{\boldsymbol{x}}_{0}^{a}\right)^{\mathrm{T}}\right]=\begin{bmatrix} \boldsymbol{P}_{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{Q}_{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{R}_{0} \end{bmatrix}$$

$$\boldsymbol{S}_{0}=\operatorname{chol}\{\boldsymbol{P}_{0}^{a}\}$$

$$\begin{cases} w_0^{(m)} = \lambda/(N+\lambda) \\ w_0^{(c)} = \lambda/(N+\lambda) + (1-\alpha^2+\eta) \quad (i=1,2,\cdots,2N) \\ w_i^{(m)} = w_i^{(c)} = 1/(2N+2\lambda) \end{cases}$$

步骤 2：样点计算

$$\boldsymbol{\chi}_{k-1}^a = \begin{bmatrix} \hat{\boldsymbol{x}}_{k-1}^a & \hat{\boldsymbol{x}}_{k-1}^a + \gamma \cdot \boldsymbol{S}_{k-1} & \hat{\boldsymbol{x}}_{k-1}^a - \gamma \cdot \boldsymbol{S}_{k-1} \end{bmatrix}$$

步骤 3：时间更新

$$\boldsymbol{\chi}_{k|k-1}^x = \boldsymbol{F}\boldsymbol{\chi}_{k-1}^x + \boldsymbol{\chi}_{k-1}^w$$

$$\hat{\boldsymbol{x}}_{k|k-1} = \sum_{i=0}^{2N} w_i^{(m)} \boldsymbol{\chi}_{i,k|k-1}^x$$

$$\boldsymbol{S}_k^- = qr\left\{\left[\sqrt{w_1^{(c)}}\left(\boldsymbol{\chi}_{1:2N,k|k-1}^x - \hat{\boldsymbol{x}}_{k|k-1}\right)\right]\right\}$$

$$\boldsymbol{S}_k^- = \text{cholupdate}\left\{\boldsymbol{S}_k^-, \left(\boldsymbol{\chi}_{0,k|k-1}^x - \hat{\boldsymbol{x}}_{k|k-1}\right), w_0^{(c)}\right\}$$

$$\boldsymbol{\mathcal{Z}}_{k|k-1} = h\left(\boldsymbol{\chi}_{k|k-1}^x\right) + \boldsymbol{\chi}_{k-1}^v$$

$$\hat{\boldsymbol{z}}_{k|k-1} = \sum_{i=0}^{2N} w_i^{(m)} \boldsymbol{\mathcal{Z}}_{i,k|k-1}$$

步骤 4：测量更新

$$\boldsymbol{D}_k^- = qr\left\{\left[\sqrt{w_1^{(c)}}\left(\boldsymbol{\mathcal{Z}}_{1:2N,k|k-1} - \hat{\boldsymbol{z}}_{k|k-1}\right)\right]\right\}$$

$$\boldsymbol{D}_k = \text{cholupdate}\left\{\boldsymbol{D}_k^-, \left(\boldsymbol{\mathcal{Z}}_{0,k|k-1} - \hat{\boldsymbol{z}}_{k|k-1}\right), w_0^{(c)}\right\}$$

$$\boldsymbol{P}_{x_k z_k} = \sum_{i=0}^{2N} w_i^{(c)} [\boldsymbol{\chi}_{i,k|k-1}^x - \hat{\boldsymbol{x}}_{k|k-1}][\boldsymbol{\mathcal{Z}}_{i,k|k-1} - \hat{\boldsymbol{z}}_{k|k-1}]^{\mathrm{T}}$$

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k(\boldsymbol{z}_k - \hat{\boldsymbol{z}}_{k|k-1})$$

$$\boldsymbol{U} = \boldsymbol{K}_k \cdot \boldsymbol{D}_k$$

$$\boldsymbol{S}_k = \text{cholupdate}\left\{\boldsymbol{S}_k^-, \boldsymbol{U}, -1\right\}$$

式中：$\boldsymbol{\chi}^a = [(\boldsymbol{\chi}^x)^{\mathrm{T}} \ (\boldsymbol{\chi}^w)^{\mathrm{T}} \ (\boldsymbol{\chi}^v)^{\mathrm{T}}]^{\mathrm{T}}$，$\boldsymbol{x}^a = [\boldsymbol{x}^{\mathrm{T}} \ \boldsymbol{w}^{\mathrm{T}} \ \boldsymbol{v}^{\mathrm{T}}]^{\mathrm{T}}$，$N = d_x + d_w + d_v$，$\lambda = \alpha^2(N+\kappa) - N$，$\gamma = \sqrt{N+\lambda}$；$\alpha$、$\kappa$、$\eta$ 为控制权值分布的参数，具体取值将在仿真中说明；$w_i^{(m)}$、$w_i^{(c)}$ 为各样点的权重；$\hat{\boldsymbol{x}}_k$ 和 $\boldsymbol{P}_k$ 为状态 $\boldsymbol{x}_k$ 的估计及协方差矩阵；qr{·} 表示矩阵的 QR 分解；chol{·}、cholupdate{·} 分别表示矩阵的 Cholesky 分解和更新。

4.3 迭代 UKF 算法

与 EKF 及其衍生算法相比，UKF 算法稳定性有了不同程度的提高。但是，在很多实际应用中（特别是在被动目标跟踪的应用条件下），受初始条件（误差较大）和可观测条件（可观测性较弱）的影响，标准的 UKF 算法存在着反应速度慢、收敛精度不高的问题。

在文献[20]中指出，由 IEKF 产生的迭代序列 $\hat{\boldsymbol{x}}_{k,0}$ 等同于 Gauss-Newton 迭代产生的序列，因此迭代过程具有全局的收敛性。理论上，IEKF 的性能必定优于 EKF 和 MVEKF，但文献[21]的研究表明，实际应用中并非总是如此。其原因之一是文献[20]中的结论是在局部线性化条件完全满足的情况下得出的，即展开点与真值非常接近。但在很多应用场合特别是被动目标跟踪应用中，由于初始误差一般都比较大，这一假设条件不能成立。原因之二是实际中测量误差不能无限小。从状态更新公式看，很明显状态量的每次修正是通过观测量完成的，因此收敛特性必然严重依赖于测量精度。文献[21]证实，IEKF 对测量误差非常敏感，故若在迭代过程中对状态量的修正完全依赖于观测值，很可能不但达不到改善滤波性能的目的，相反会导致严重的性能恶化。原因之三是尽管 Gauss-Newton 方法具有全局的收敛性，但不能确保似然的增加[22]。此外，门限 V_{th} 的选择对迭代算法的成败至关重要，但如何选择合适的 V_{th} 却是不好解决的问题。

受 IEKF 算法的启发，同时考虑到 UKF 本身所具有的优势，一个自然的想法是如果将迭代算法与 UKF 相结合，可能会带来滤波器性能的改善。但是由于 UKF 是基于样点变换的方法，如何使迭代收敛是个问题；同时，鉴于传统 IEKF 存在的潜在问题，必须采取合适的迭代策略使其扬长避短，真正发挥迭代算法的优势。在接下来提出的 IUKF 算法中将解决这些问题，算法中采用了不同于 IEKF 的迭代策略，其主要流程可概括如下[23,24]。

Alg（4.3）：Iterated Unscented Kalman Filtering

步骤 1：在 k（$k \geqslant 1$）时刻，通过式（4-22）～式（4-32）所示的 UKF 算法得到状态量的估计 $\hat{\boldsymbol{x}}_k$ 和相应的协方差矩阵 $\boldsymbol{P}_k$。

步骤 2：令 $\hat{\boldsymbol{x}}_{k,0} = \hat{\boldsymbol{x}}_{k|k-1}$，$\boldsymbol{P}_{k,0} = \boldsymbol{P}_{k|k-1}$ 且 $\hat{\boldsymbol{x}}_{k,1} = \hat{\boldsymbol{x}}_k$，$\boldsymbol{P}_{k,1} = \boldsymbol{P}_k$。同时令 $g = 1$，$j = 2$。

步骤 3：根据新的协方差矩阵产生样点。

$$\boldsymbol{\chi}_{i,j} = [\hat{\boldsymbol{x}}^{*a}_{k,j-1} \quad \hat{\boldsymbol{x}}^{*a}_{k,j-1} \pm \sqrt{(L^{*a} + \kappa)\boldsymbol{P}^{*a}_{k,j-1}}\,]_i \tag{4-33}$$

式中

$$\hat{\boldsymbol{x}}^{*a}_{k,j-1} = [\hat{\boldsymbol{x}}^{\mathrm{T}}_{k,j-1} \ \ \boldsymbol{0}]^{\mathrm{T}}, \quad \boldsymbol{P}^{*a}_{k,j-1} = \begin{bmatrix} \boldsymbol{P}_{k,j-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_k \end{bmatrix}$$

步骤 4：对式（4-24）～式（4-32）重新计算。

$$\hat{\boldsymbol{x}}_{k|k-1,j} = \sum_{i=0}^{2L_a^*} w_i^{(m)} \boldsymbol{\chi}_{i,j} \tag{4-34}$$

$$\boldsymbol{\mathcal{Y}}_{i,j} = h(\boldsymbol{\chi}^x_{i,j}, \boldsymbol{\chi}^v_{i,j}) \tag{4-35}$$

$$\hat{\boldsymbol{y}}_{k|k-1,j} = \sum_{i=0}^{2L_a^*} w_i^{(m)} \boldsymbol{\mathcal{Y}}_{i,j} \tag{4-36}$$

$$\boldsymbol{P}_{yy,k,j} = \sum_{i=0}^{2L_a^*} w_i^{(c)} [\boldsymbol{\mathcal{Y}}_{i,j} - \hat{\boldsymbol{y}}_{k|k-1,j}][\boldsymbol{\mathcal{Y}}_{i,j} - \hat{\boldsymbol{y}}_{k|k-1,j}]^{\mathrm{T}} \tag{4-37}$$

$$\boldsymbol{P}_{xy,k,j} = \sum_{i=0}^{2L_a^*} w_i^{(c)} [\boldsymbol{\chi}^x_{i,j} - \hat{\boldsymbol{x}}_{k|k-1,j}][\boldsymbol{\mathcal{Y}}_{i,j} - \hat{\boldsymbol{y}}_{k|k-1,j}]^{\mathrm{T}} \tag{4-38}$$

$$\boldsymbol{K}_{k,j} = \boldsymbol{P}_{xy,k,j} \boldsymbol{P}^{-1}_{yy,k,j} \tag{4-39}$$

$$\hat{\boldsymbol{x}}_{k,j} = \hat{\boldsymbol{x}}_{k|k-1,j} + g \cdot \boldsymbol{K}_{k,j}(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_{k|k-1,j}) \tag{4-40}$$

$$\boldsymbol{P}_{k,j} = \boldsymbol{P}_{k,j-1} - \boldsymbol{K}_{k,j}\boldsymbol{P}_{yy,k,j}\boldsymbol{K}_{k,j}^{\mathrm{T}} \tag{4-41}$$

式中：下标 j 表示第 j 次迭代；$\mathcal{Y}_{i,j}$ 表示变换点集 $\mathcal{Y}_j$ 的第 i 个分量。

步骤 5：定义如下 3 个式子，即

① $\hat{\boldsymbol{y}}_{k,j} = h(\hat{\boldsymbol{x}}_{k,j})$

② $\tilde{\boldsymbol{x}}_{k,j} = \hat{\boldsymbol{x}}_{k,j} - \hat{\boldsymbol{x}}_{k,j-1}$

③ $\tilde{\boldsymbol{y}}_{k,j} = \boldsymbol{y}_k - \hat{\boldsymbol{y}}_{k,j}$

步骤 6：若下式成立（见附录 C）

$$\tilde{\boldsymbol{x}}_{k,j}^{\mathrm{T}}\boldsymbol{P}_{k,j-1}^{-1}\tilde{\boldsymbol{x}}_{k,j} + \tilde{\boldsymbol{y}}_{k,j}^{\mathrm{T}}\boldsymbol{R}_k^{-1}\tilde{\boldsymbol{y}}_{k,j} < \tilde{\boldsymbol{y}}_{k,j-1}^{\mathrm{T}}\boldsymbol{R}_k^{-1}\tilde{\boldsymbol{y}}_{k,j-1} \tag{4-42}$$

且 $j \leqslant N$，则令 $g = \eta \cdot g$，$j = j+1$ 返回步骤 3；否则执行步骤 7。

步骤 7：若式（4-42）不成立或 j 过大（$j > N$）则迭代终止，并令 $\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_{k,j}$，$\boldsymbol{P}_k = \boldsymbol{P}_{k,j}$。

下面说明上述迭代过程是收敛的。首先由式（4-41）知，对于正定矩阵 $\boldsymbol{P}_{k,j}$、$\boldsymbol{P}_{k,j-1}$ 和 $\boldsymbol{P}_{yy,k,j}$，若假设 $\lim\limits_{j\to\infty}\boldsymbol{K}_{k,j} \neq \boldsymbol{0}$，则根据式（4-41）可知对任意 $j = 1,2,\cdots < \infty$ 都有 $\boldsymbol{P}_{k,j} < \boldsymbol{P}_{k,j-1}$ ① 成立。同时，由于矩阵 $\boldsymbol{P}_{k,j}$ 中的每一个元素是有界的，由此得（见附录 D 证明）

$$\lim_{j\to\infty}\boldsymbol{P}_{k,j} = \lim_{j\to\infty}\boldsymbol{P}_{k,j-1} \tag{4-43}$$

显然在式（4-43）成立的前提下，由式（4-41）可推知 $\lim\limits_{j\to\infty}\boldsymbol{K}_{k,j} = \boldsymbol{0}$，这与前面的假设条件 $\lim\limits_{j\to\infty}\boldsymbol{K}_{k,j} \neq \boldsymbol{0}$ 相矛盾。故原来的假设不成立，唯一的可能是 $\lim\limits_{j\to\infty}\boldsymbol{K}_{k,j} = \boldsymbol{0}$。现假定当 $j > N$ 时有 $\boldsymbol{K}_{k,j} \to \boldsymbol{0}$，则由式（4-40）和式（4-41）可得 $\hat{\boldsymbol{x}}_{k,j} \to \hat{\boldsymbol{x}}_{k|k-1,j} = \hat{\boldsymbol{x}}_{k,j-1}$ 且 $\boldsymbol{P}_{k,j} \to \boldsymbol{P}_{k,j-1}$，这就说明了迭代过程是收敛的。

根据式（4-40）可知，第 N 次迭代的结果为 $\hat{\boldsymbol{x}}_{k,N} = \hat{\boldsymbol{x}}_{k|k-1,N} + \eta^{N-1} \cdot \boldsymbol{K}_{k,N}(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_{k|k-1,N})$。若衰减因子 η 的取值为 $0 < \eta < 1$，则当 N 充分大时，由于 $\eta^{N-1} \to 0$，有 $\hat{\boldsymbol{x}}_{k,N} \to \hat{\boldsymbol{x}}_{k|k-1,N} = \hat{\boldsymbol{x}}_{k,N-1}$。由此同样可知，迭代过程必然会收敛到某一解，且收敛速度受 η 的制约。

与传统的 IEKF 算法相比，IUKF 在三个方面有明显的区别：首先在步骤 6 中，式（4-42）确保了迭代过程朝似然增加的方向进行，即迭代过程逼近最大似然意义下的最优解（有关证明见附录 C）；其次是步长调整策略，衰减因子 η（$0 < \eta \leqslant 1$）用于削弱新近一次观测对预测状态（$\hat{\boldsymbol{x}}_{k|k-1,j}$）的修正影响，这将使得相邻两次迭代结果（$\hat{\boldsymbol{x}}_{k,j}$ 和 $\hat{\boldsymbol{x}}_{k,j-1}$）随迭代的进行越来越接近，从而加速收敛过程；最后，在每一次迭代中，新的样点通过式（4-33）得到，这意味着样点的生成是由最近一次更新的协方差矩阵来引导的，使得样点逐步向真实的状态收敛。

就算法复杂度而言，由文献[19]可知对于维数为 L 的状态估计问题，UKF 的算法复杂度与 EKF 相当，均为 $O(L^3)$。类似地，在任意时刻 k，若进行 N 次迭代，则 IUKF 的复杂度约为 $O(NL^3)$。由此可见，只要 N 不过大，IUKF 运算量与 UKF 基本在同一量级，实现起来并不困难。

---

① 对于正定矩阵 $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{m\times n}$，记 $\boldsymbol{A} < \boldsymbol{B}$，其含义是对任意矢量 $\boldsymbol{a} \in \mathbb{R}^n$ 且 $\boldsymbol{a} \neq \boldsymbol{0}$，则有 $\boldsymbol{a}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{a} < \boldsymbol{a}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{a}$。

4.4 简化 UKF 算法

在标准的 UKF 算法中，通常将系统状态、过程噪声和观测噪声扩展成新的状态量来处理，这在维数较高的情况下算法复杂度将会显著增加；此外，标准的 UKF 算法同时将 UT 应用于状态方程和观测方程，不管它们是否为线性。然而，在大多数应用中，噪声是加性的，并且状态方程和观测方程并不都是非线性的，对这类系统，可以在不影响滤波器性能的前提下对 UKF 算法进行简化、降低其运算复杂度。本节主要针对目标跟踪的实际应用背景，根据系统的特点提出一种简化的 UKF（SUKF）算法，以降低运算开销，便于实时应用。

假设状态量、过程噪声和观测噪声的维数分别为 L_x、L_w 和 L_v，则增广后得到的状态维数为 $L_a = L_x + L_w + L_v$，样点个数为 $2L_a + 1$，算法的复杂度为 $O(L_a^3)$。相比之下，经典 EKF 的算法复杂度为 $O(L_x^3)$。显然，标准 UKF 算法的复杂度与增广状态量的维数呈三次方递增，当 L_x、L_w 和 L_v 的维数都比较高时，运算量的增加非常显著，这对实时应用是不利的。

事实上，对于大多数应用背景，都具有这样的特点：①过程噪声和观测噪声至少有一项是加性的；②式（4-16）和式（4-17）并非都是非线性的。对于这一类问题，可以在保证滤波器性能的基础上对标准的 UKF 算法进行简化，从而达到降低运算量的目的。此时，对于加性噪声，无需对状态量进行增广，可以不同程度地减小运算量；而对于线性方程，理论上 KF 是最优的，因此可以直接用 KF 进行运算，而不必用多个样点来近似。为表述方便，这里把这一类系统所对应的 UKF 算法称为 SUKF（Simplified Unscented Kalman Filtering）算法[25]。

对于式（4-16）和式（4-17）所描述的系统，根据前面的讨论，可以分成表 4.1 所列的几种不同的形式。显然，形式①最为复杂，对应于标准的 UKF 算法；而形式⑨最为简单，退化为 KF 算法，其他几种形式都可以用 SUKF 算法来处理。

表 4.1 几种不同的系统方程形式

$\boldsymbol{x}_k$ \ $\boldsymbol{y}_k$	$h(\boldsymbol{x}_k,\boldsymbol{v}_k)$	$h(\boldsymbol{x}_k)+\boldsymbol{v}_k$	$\boldsymbol{H}\boldsymbol{x}_k+\boldsymbol{v}_k$
$f(\boldsymbol{x}_k,\boldsymbol{w}_k)$	①	②	③
$f(\boldsymbol{x}_k)+\boldsymbol{w}_k$	④	⑤	⑥
$\boldsymbol{F}\boldsymbol{x}_k+\boldsymbol{w}_k$	⑦	⑧	⑨

形式⑧所描述的一类重要系统为

$$\boldsymbol{x}_k = \boldsymbol{F}\boldsymbol{x}_{k-1} + \boldsymbol{w}_k \tag{4-44}$$

$$\boldsymbol{y}_k = h(\boldsymbol{x}_k) + \boldsymbol{v}_k \tag{4-45}$$

在这种条件下，状态方程是线性的，而观测方程是非线性的，且两种噪声都是加性的高斯噪声。由前面的讨论知，用 SUKF 算法进行处理时，不对状态量进行扩展，同时对线性部分直接应用 KF 进行计算，由此得到的 SUKF 算法可表述如下。

Alg（4.4）：Simplified Unscented Kalman Filtering

步骤 1：初始化

$$\hat{x}_0 = E[x_0] \tag{4-46}$$

$$P_0 = E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)]^{\mathrm{T}} \tag{4-47}$$

步骤 2：计算预测值和方差

对于 $k \in \{1, \cdots, \infty\}$，有

$$\hat{x}_{k|k-1} = F\hat{x}_{k-1} \tag{4-48}$$

$$P_{xx} = FP_{k-1}F^{\mathrm{T}} + GQ_kG^{\mathrm{T}} \tag{4-49}$$

步骤 3：时间更新

$$x^{(i)} = [\hat{x}_{k|k-1} \quad \hat{x}_{k|k-1} + \sqrt{(L+\kappa)P_{xx}} \quad \hat{x}_{k|k-1} - \sqrt{(L+\kappa)P_{xx}}\,]_i \tag{4-50}$$

$$\hat{y}_{k|k-1} = \sum_{i=0}^{2L} w_i^{(m)} h(x^{(i)}) \tag{4-51}$$

式中：$i = 0,1,\cdots,2L$，L 为状态量维数。

步骤 4：测量更新

$$P_{yy} = \sum_{i=0}^{2L} w_i^{(c)} [h(x^{(i)}) - \hat{y}_{k|k-1}][h(x^{(i)}) - \hat{y}_{k|k-1}]^{\mathrm{T}} + R_k \tag{4-52}$$

$$P_{xy} = \sum_{i=0}^{2L} w_i^{(c)} [x^{(i)} - \hat{x}_{k|k-1}][h(x^{(i)}) - \hat{y}_{k|k-1}]^{\mathrm{T}} \tag{4-53}$$

$$\hat{x}_k = \hat{x}_{k|k-1} + P_{xy}P_{yy}^{-1}(y_k - \hat{y}_{k|k-1}) \tag{4-54}$$

$$P_k = P_{xx} - P_{xy}P_{yy}^{-1}P_{xy}^{\mathrm{T}} \tag{4-55}$$

步骤 5：重复步骤 2～步骤 4。

以上即为线性状态空间、非线性观测模型对应的 SUKF 算法的流程，可广泛应用于一般的目标跟踪应用场合。对于其他方面的应用，可根据实际的条件和背景得到相应的算法。

4.5 联合估计 UKF 算法

前面讨论的是 UKF 用于状态估计的问题，而状态量本身是一个时变的随机变量。在实际应用中，通常会碰到下式所描述的问题，即

$$x_k = f(x_{k-1}, u) + w_k \tag{4-56}$$

$$y_k = h(x_k) + v_k \tag{4-57}$$

即状态变量 x_k 中包含某一未知的确定性常量 u（称为参量或参数），该常量可能是匀加速目标中的加速度输入、转弯机动中的角速度，或者其他类似的参量。此时，只有得到状态量和参量的完整信息才能获得准确的状态估计。在 1.3.3 节介绍线性系统中的滤波问题时已经提到，在解决这类问题时，通常有两种途径，分别称为双重估计（Dual Estimation）和联合估计（Joint Estimation）。前者利用两次滤波估计，第一次估计状态变量，第二次估计未知参数；而后者则是将未知参数与状态变量扩展成新的状态空间 $x_k^a = [x_k^{\mathrm{T}} \; u_k^{\mathrm{T}}]^{\mathrm{T}}$，并利用滤波算法同时估计求解。

由于联合估计将参数和状态合成为一个扩展状态量，因此能对参数和状态之间的耦合量（这里指互协方差）

$$E[(\boldsymbol{x}_k^a - E[\boldsymbol{x}_k^a])(\boldsymbol{x}_k^a - E[\boldsymbol{x}_k^a])^{\mathrm{T}}] = \begin{bmatrix} \boldsymbol{P}_{x_k x_k} & \boldsymbol{P}_{x_k u_k} \\ \boldsymbol{P}_{u_k x_k} & \boldsymbol{P}_{u_k u_k} \end{bmatrix} \tag{4-58}$$

进行准确建模，即式（4-58）考虑了状态和参数间的相关信息。而对于双重估计，由于将参数和状态分开来独立考虑，在考虑某个量时将另外一个量视为确定的常量（非随机变量），即隐含地将上式中副对角线上的元素视为零来处理。由于联合估计更接近于真实的模型，因此能得到更好的性能。

基于这样的推理，得到的 JEUKF 算法[26]流程可概括如下。

Alg（4.5）：Joint Estimation Unscented Kalman Filtering

步骤 1：初始化

$$\hat{\boldsymbol{x}}_0 = E[\boldsymbol{x}_0] \tag{4-59}$$

$$\boldsymbol{P}_0 = E[(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)^{\mathrm{T}}] \tag{4-60}$$

步骤 2：计算样点

$$\begin{cases} \chi_{0,k-1} = \hat{\boldsymbol{x}}_{k-1}, & w_0 = \kappa/(L+\kappa) \\ \chi_{i,k-1} = \hat{\boldsymbol{x}}_{k-1} + (\sqrt{(L+\kappa)\boldsymbol{P}_{k-1}})_i, & w_i = 0.5/(L+\kappa) \\ \chi_{i+L,k-1} = \hat{\boldsymbol{x}}_{k-1} - (\sqrt{(L+\kappa)\boldsymbol{P}_{k-1}})_i, & w_{i+L} = 0.5/(L+\kappa) \end{cases} \tag{4-61}$$

式中：$i = 1, 2, \cdots, L$ 。L 为扩展状态量 $\boldsymbol{x}$ 的维数；参量 κ 用于控制协方差阵。

步骤 3：时间更新

$$\chi_{i,k|k-1} = f(\chi_{i,k-1}, \boldsymbol{u}_{i,k-1}) \tag{4-62}$$

$$\hat{\boldsymbol{x}}_{k|k-1} = \sum_{i=0}^{2L} w_i \chi_{i,k|k-1} \tag{4-63}$$

$$\boldsymbol{P}_{k|k-1} = \sum_{i=0}^{2L} w_i [\chi_{i,k|k-1} - \hat{\boldsymbol{x}}_{k|k-1}][\chi_{i,k|k-1} - \hat{\boldsymbol{x}}_{k|k-1}]^{\mathrm{T}} + \boldsymbol{Q}_k \tag{4-64}$$

$$\mathcal{Y}_{i,k|k-1} = h(\chi_{i,k|k-1}) \tag{4-65}$$

$$\hat{\boldsymbol{y}}_{k|k-1} = \sum_{i=0}^{2L} w_i \mathcal{Y}_{i,k|k-1} \tag{4-66}$$

步骤 4：测量更新

$$\boldsymbol{P}_{yy} = \sum_{i=0}^{2L} w_i [\mathcal{Y}_{i,k|k-1} - \hat{\boldsymbol{y}}_{k|k-1}][\mathcal{Y}_{i,k|k-1} - \hat{\boldsymbol{y}}_{k|k-1}]^{\mathrm{T}} + \boldsymbol{R}_k \tag{4-67}$$

$$\boldsymbol{P}_{xy} = \sum_{i=0}^{2L} w_i [\chi_{i,k|k-1} - \hat{\boldsymbol{x}}_{k|k-1}][\mathcal{Y}_{i,k|k-1} - \hat{\boldsymbol{y}}_{k|k-1}]^{\mathrm{T}} \tag{4-68}$$

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_{k|k-1}) \tag{4-69}$$

$$\boldsymbol{P}_k = \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k \boldsymbol{P}_{yy} \boldsymbol{K}_k^{\mathrm{T}} \tag{4-70}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_{xy} \boldsymbol{P}_{yy}^{-1} \tag{4-71}$$

4.6 应用实例

4.6.1 被动目标跟踪

考虑 x-y 二维平面内被动跟踪问题，目标状态方程为

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}\boldsymbol{x}_k + \boldsymbol{G}\boldsymbol{w}_k \tag{4-72}$$

式中：$\boldsymbol{F} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}$；$\boldsymbol{G} = \begin{bmatrix} 0.5T^2 & 0 \\ T & 0 \\ 0 & 0.5T^2 \\ 0 & T \end{bmatrix}$；$\boldsymbol{x}_k = [x_k \ \dot{x}_k \ y_k \ \dot{y}_k]^{\mathrm{T}}$；$\boldsymbol{w}_k = [w_x \ w_y]^{\mathrm{T}}$ 为扰动噪声，且 $E[\boldsymbol{w}_k] = \boldsymbol{0}$，$\mathrm{cov}[\boldsymbol{w}_k] = \boldsymbol{Q}_k \delta_{kj}$。

观测量包括角度 φ、角度变化率 $\dot{\varphi}$ 和频率变化率 $\dot{f}_d$，观测方程表示为

$$\boldsymbol{y}_k = \begin{bmatrix} \varphi_k \\ \dot{\varphi}_k \\ \dot{f}_{dk} \end{bmatrix} = \begin{bmatrix} \arctan\left(\dfrac{x_k}{y_k}\right) \\ \dfrac{\dot{x}_k y_k - x_k \dot{y}_k}{x_k^2 + y_k^2} \\ -\dfrac{(\dot{x}_k y_k - x_k \dot{y}_k)^2}{\lambda (x_k^2 + y_k^2)^{3/2}} \end{bmatrix} + \begin{bmatrix} v_{\varphi,k} \\ v_{\dot{\varphi},k} \\ v_{\dot{f}_d,k} \end{bmatrix} \triangleq h(\boldsymbol{x}_k) + \boldsymbol{v}_k \tag{4-73}$$

式中：λ 为信号波长；$v_{\varphi,k}$、$v_{\dot{\varphi},k}$ 和 $v_{\dot{f}_d,k}$ 为观测噪声且相互独立，其方差阵 $\boldsymbol{R}_k = \mathrm{diag}\{\sigma_{\varphi}^2, \sigma_{\dot{\varphi}}^2, \sigma_{\dot{f}_d}^2\}$。

假定观测站位于坐标原点，目标在平面内作近匀速运动，其初始状态为

$$\boldsymbol{x}_0 = [120\mathrm{km} \ -250\mathrm{m/s} \ 80\mathrm{km} \ 100\mathrm{m/s}]^{\mathrm{T}} \tag{4-74}$$

目标的初始位置估计由测距公式

$$r = \lambda \dot{f}_d / \dot{\beta}^2 \tag{4-75}$$

综合角度信息得到。

在仿真中，过程噪声协方差矩阵 $\boldsymbol{Q}_k = \mathrm{diag}\{(3\mathrm{m/s^2})^2, (2\mathrm{m/s^2})^2\}$。对不同观测精度（{2e–3rad，1e–4rad/s，0.5Hz/s}；{5e-3rad，2e–4rad/s，1Hz/s}；{0.01rad，5e–4rad/s，2Hz/s}）条件下的算法性能进行测试，性能指标用相对距离误差（RRE）来测度，其定义为

$$\mathrm{RRE} = \frac{\sqrt{(x_{\mathrm{true}} - \hat{x})^2 + (y_{\mathrm{true}} - \hat{y})^2}}{\sqrt{x_{\mathrm{true}}^2 + y_{\mathrm{true}}^2}} \times 100\% \tag{4-76}$$

式中：$(x_{\mathrm{true}}, y_{\mathrm{true}})$ 为真实目标位置；$(\hat{x}, \hat{y})$ 表示位置估值。

在每一组仿真中取 100 次蒙特卡罗实验，其结果见表 4.2[①]。由表中的结果可以看出，在高精度观测条件下，几种跟踪算法都能较好地收敛。但是，随着测量误差的增大，各

① 在仿真 1~仿真 3 中，测量误差分别为{2e–3rad，1e–4rad/s，0.5Hz/s}，{5e–3rad，2e–4rad/s，1Hz/s}和{1e–2rad，5e–4rad/s，2Hz/s}；在每次试验中，跟踪时间为 120s；若在跟踪结束时刻 RRE <15%则本次跟踪视为收敛，否则视为发散；IUKF 和 $\mathrm{IEKF_2}$ 采用同一迭代策略，其最大迭代次数 N=5，衰减因子 η=0.85；而 $\mathrm{IEKF_1}$ 则采用传统迭代策略，其门限为 V_{th}=100。

种算法的性能恶化程度却大不相同。相比之下，EKF 算法由于线性化的影响对测量误差更加敏感。从表中可以看出，迭代型 EKF（$IEKF_1$ 和 $IEKF_2$）的性能很大程度上受迭代策略的影响，在收敛性准则选择合适的情况下 $IEKF_2$ 总能比 EKF 和 MVEKF 得到更好的估计性能，这一点与前面的理论分析是一致的。这是因为，本质上 EKF 和 MVEKF 为迭代型 EKF 的两种特殊形式，对应于无迭代和 1 次迭代。同样，当迭代门限选择不当时，可能会导致算法性能严重恶化，如表中的 $IEKF_1$。

表 4.2　跟踪算法的鲁棒性比较

100 次试验中跟踪收敛的次数			
Algorithms	Simulation 1	Simulation 2	Simulation 3
EKF	100	76	43
$IEKF_1$	100	73	38
$IEKF_2$	100	88	68
MVEKF	100	85	49
UKF	100	91	74
IUKF	100	100	88

仿真中发现，文中的收敛性准则一般经 2 次～3 次就能满足，这意味着 IUKF 的平均运算开销约为 UKF 的 2 倍～3 倍。

在仿真中，参数 κ 默认地设为 0，但实际上只要保证所生成的样点有良好的数值稳定性（权值 w_i 不为负数），算法的性能对该值并不敏感；同样 η 的选择也可比较随意，当 $0.5<\eta\leqslant 1$，对算法的性能影响不大，但若 η 选择不当（如 $\eta>1$）可能会导致滤波器不稳定。

为了提供更直观的性能比较，图 4.2 示出了中等观测精度（仿真 2）条件下几种算法的平均 RRE。需要指出的是，图 4.2 中的结果反映的是蒙特卡罗试验的统计性能，且已剔除了跟踪发散的试验（RRE >15%）。

如图 4.2 所示，UKF 无论在收敛时间还是跟踪误差方面都要优于 EKF，在统计平均的意义下 MVEKF 看似比 UKF 跟踪误差要小，但不能简单断言 MVEKF 的性能要优于 UKF，这是因为，在同一组仿真中 MVEKF 剔除了更多不收敛的试验结果。从图中可以看出，与非迭代型滤波器相比，迭代型滤波器通常有更快的收敛速度，这一特点使得迭代型滤波器更具优势，特别在初始误差较大的情况下更是如此。很明显，IUKF 无论在收敛速度和跟踪精度上都要优于其他几种算法（EKF，$IEKF_1$，$IEKF_2$，UKF 和 MVEKF），尽管这些算法已经剔除了不收敛的跟踪结果。

在实际应用中，通常更关心的是单次试验中算法的稳定性，从这一意义来说，IUKF 的优势更加明显。

为进一步分析所提 IUKF 算法的稳定性，图 4.3 给出了状态估计标准偏差和跟踪误差的比较示例。不同坐标轴的跟踪误差定义为

$$e_x = x_{\text{true}} - \hat{x}\,,\quad e_y = y_{\text{true}} - \hat{y} \tag{4-77}$$

相应的标准偏差表示为

$$\sigma_x = \sqrt{p_{11}}\ ,\quad \sigma_y = \sqrt{p_{33}} \tag{4-78}$$

式中：p_{ii} 表示跟踪误差协方差矩阵 $\boldsymbol{P}$ 的主对角元素。

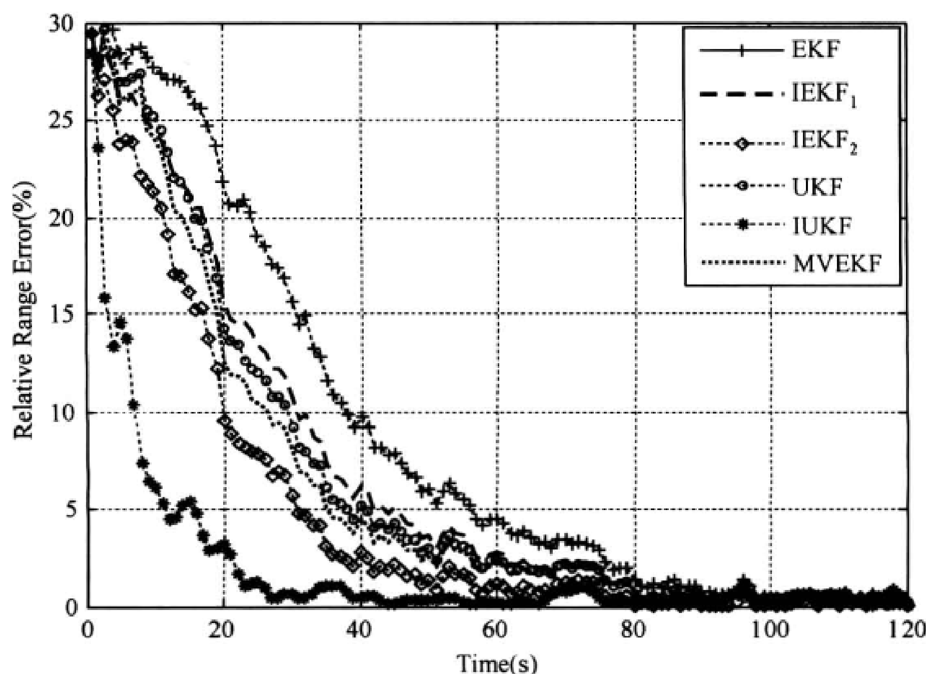


图 4.2　几种算法的跟踪结果比较

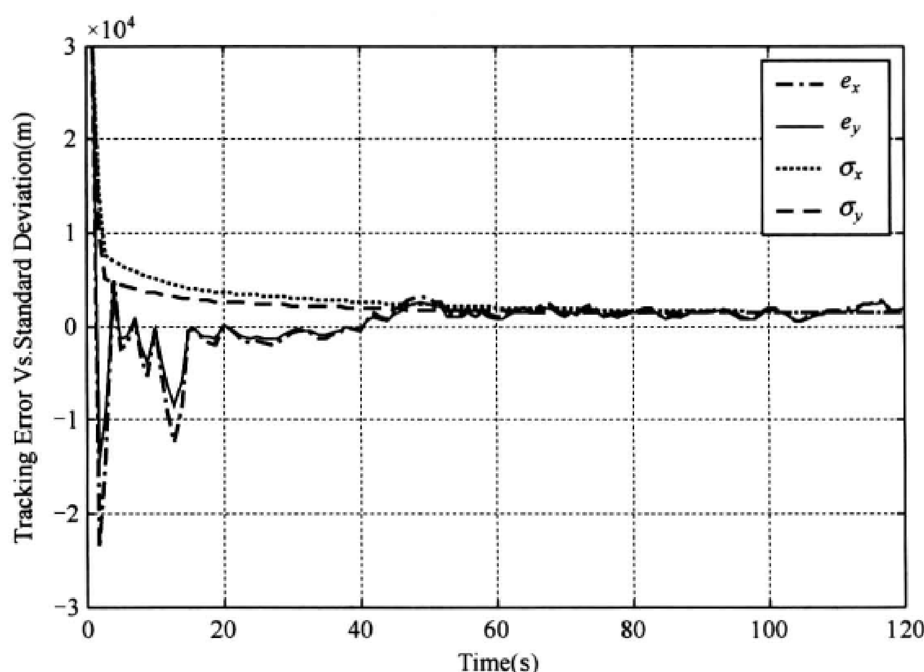


图 4.3　跟踪误差和标准偏差比较

从图 4.3 可以看出，当滤波器收敛之后，其状态估计的标准偏差与跟踪误差能较好地吻合。这就意味着，标准偏差能准确描述估计误差。由此可以推知，一旦标准偏差达到稳定状态，跟踪误差也相应地趋于稳定（即误差不会有大的起伏）。

为进一步验证算法性能，将 IUKF 用于外场实测数据处理。试验场景如图 4.4 所示，载有雷达信号源的小车从某跑道的一端运动到另一端，观测平台固定不动，它与辐射源运动轨迹的垂直距离约为 520m。通过雷达接收信号、参数估计等手段获得目标辐射源的波达角、角度变化率和频率变化率，综合利用 φ、$\dot{\varphi}$ 和 $\dot{f}_d$ 进行定位与跟踪。在试验中，

采用差分 GPS 提供的高精度位置信息作为定位与跟踪结果的参考数据。

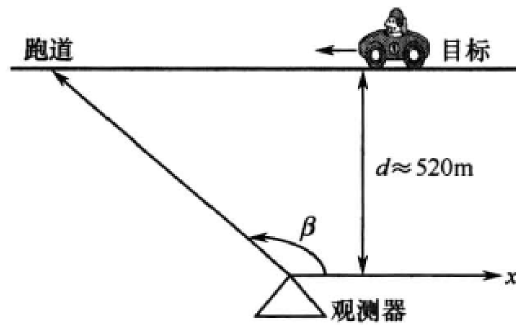


图 4.4 目标—观测器相对几何示意

试验中，辐射源以约 15/m 的速度从右向左运动，统计结果表明其参数测量精度为{0.36mrad，2.5mrad/s，2.8Hz/s}。采用几种不同算法[①]（EKF，$IEKF_1$，$IEKF_2$，UKF，IUKF 和 MVEKF）分别对目标进行跟踪，其结果如图 4.5 和图 4.6 所示。

由图 4.5 和图 4.6 可以清楚地看出，迭代型滤波器（$IEKF_1$，$IEKF_2$ 和 IUKF）能比非迭代型滤波器（EKF 和 UKF）对新的观测做出更快的反应，也正因为此，在测量精度足够高的情况下，迭代型滤波器的收敛速度也相对较快。很明显，UKF 和 MVEKF 都比经典 EKF 有更好的性能，这是因为这两种算法对状态均值和协方差的近似精度比 EKF 要高。试验结果同样表明，对于迭代型 EKF，其迭代策略对结果产生重要的影响，因此合适的终止准则和门限选择是提高滤波器性能的关键。

与前面的仿真结果一致的是，IUKF 无论在收敛速度还是跟踪误差方面都要优于其他几种算法。IUKF 的这一显著优势主要来自于两个方面：一是 UKF 比 EKF 的估计精度要高，这一点在仿真和试验结果中都得以体现；二是因为迭代型 UKF 比标准 UKF 有着更快的反应速度，从而能够缩短收敛时间。

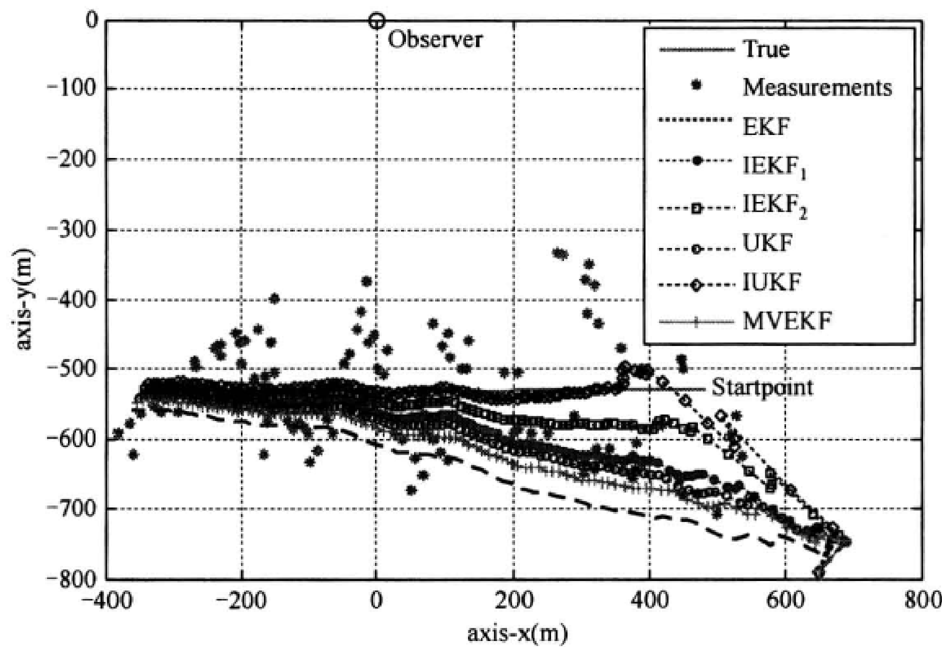


图 4.5 几种滤波器跟踪结果比较

① 本试验中，IUKF 和 $IEKF_2$ 采用同一迭代策略且 N=5，η=0.8；$IEKF_1$ 采用传统迭代算法，终止门限 V_{th}=50。

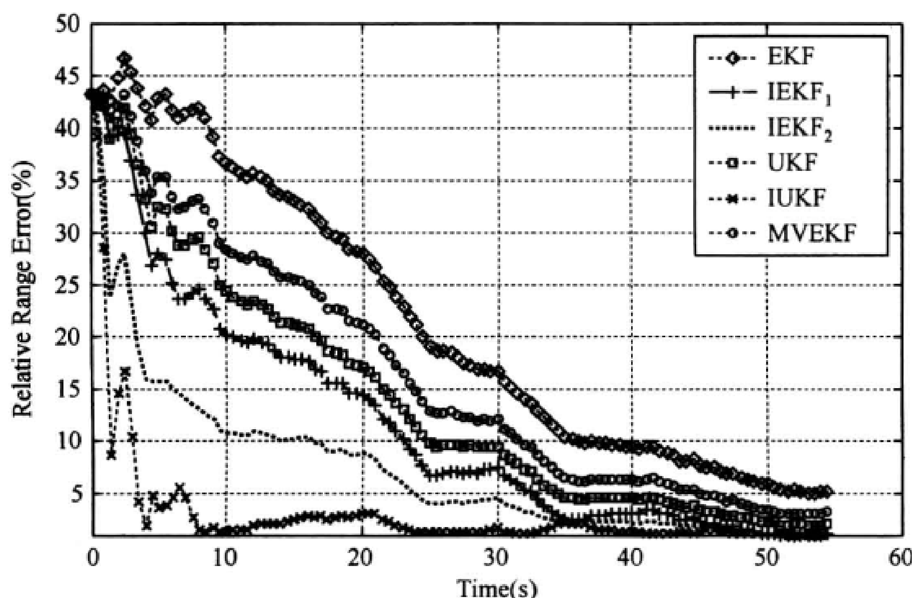


图 4.6 相对距离误差比较

4.6.2 转弯目标跟踪

本实例主要针对目标转弯机动的运动特点，提出并验证一种利用单个模型结合联合估计 UKF 算法（JEUKF）跟踪机动目标的能力。

考虑二维平面的目标运动问题，直角坐标系下匀速运动（CV）目标的离散状态方程可表示为

$$\boldsymbol{x}_k = \boldsymbol{F}_{\mathrm{CV}}\boldsymbol{x}_{k-1} + \boldsymbol{\varGamma}\boldsymbol{w}_k \tag{4-79}$$

式中：$\boldsymbol{x}_k = [x_k\ \dot{x}_k\ y_k\ \dot{y}_k]^{\mathrm{T}}$ 为目标状态；$\boldsymbol{w}_k = [w_{x_k}\ \ w_{y_k}]^{\mathrm{T}}$ 为零均值扰动噪声，其协方差矩阵为 $\boldsymbol{Q}_k$；T 为采样间隔，且有

$$\boldsymbol{F}_{\mathrm{CV}} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix},\quad \boldsymbol{\varGamma} = \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & T^2/2 \\ 0 & T \end{bmatrix}$$

当目标做转弯机动时，若在某段时间内，转弯率（角速度）保持恒定不变，则称为恒转弯（CT）运动或圆周运动。此时，若角速度 ω 已知，则相应的运动方程为

$$\boldsymbol{x}_k = \boldsymbol{F}_{\mathrm{CT}}\boldsymbol{x}_{k-1} + \boldsymbol{\varGamma}\boldsymbol{w}_k \tag{4-80}$$

式中

$$\boldsymbol{F}_{\mathrm{CT}} = \begin{bmatrix} 1 & \dfrac{\sin(\omega T)}{\omega} & 0 & \dfrac{\cos(\omega T)-1}{\omega} \\ 0 & \cos(\omega T) & 0 & -\sin(\omega T) \\ 0 & \dfrac{1-\cos(\omega T)}{\omega} & 1 & \dfrac{\sin(\omega T)}{\omega} \\ 0 & \sin(\omega T) & 0 & \cos(\omega T) \end{bmatrix}$$

其他量的含义与式（4-79）相同。

显然，对于非合作目标，很难确知 ω 的先验信息，故只能将 ω 作为一个未知的参数来估计。此时，扩展的状态量为 $\boldsymbol{x}_k = [x_k\ \dot{x}_k\ y_k\ \dot{y}_k\ \ \omega_k]^{\mathrm{T}}$，于是运动模型变为

$$\boldsymbol{x}_k = \boldsymbol{F}^a \boldsymbol{x}_{k-1} + \boldsymbol{\Gamma}^a \boldsymbol{w}_k \tag{4-81}$$

式中

$$\boldsymbol{F}^a = \mathrm{diag}\{\boldsymbol{F}_{\mathrm{CT}}, \omega\}, \quad \boldsymbol{\Gamma}^a = \mathrm{diag}\{\boldsymbol{\Gamma}, 1\}, \quad \boldsymbol{w}_k = [w_{x_k}\ w_{y_k}\ w_{\omega}]^{\mathrm{T}}$$

对于机动目标跟踪，常用的方法主要有输入估计（IE）、变维滤波（VDF）和交互多模（IMM）等。IE 算法和 VDF 算法需要对目标机动进行检测，故存在检测滞后问题，跟踪误差较大；IMM 算法则采用若干模型并行工作，虽然不存在机动检测滞后的问题，但由于需要同时运行多个滤波器，增加了运算开销，且多个模型之间的竞争会降低算法的跟踪性能。有关多模型滤波问题将在第 12 章进行详细阐述。

由式(4-81)知，要对目标进行有效跟踪，需要同时对目标状态（包括位置、速度）和机动参数（角速度 ω ）进行准确估计，属典型的联合估计（JE）问题。同时注意到在式（4-80）中，当 ω 较小（ $\omega \to 0$ ）时，有

$$\lim_{\omega \to 0} \boldsymbol{F}_{\mathrm{CT}} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} = \boldsymbol{F}_{\mathrm{CV}} \tag{4-82}$$

显然，式（4-82）正是匀速运动所对应的状态转移矩阵。由此可知，只要能准确地估计出角速率 ω ，就可以用单个模型跟踪目标的运动，与文献[1]中采用 CV 模型和 CT 模型组成的 IMM 算法相比，可以在保证跟踪性能的条件下大大简化算法的复杂度。

仿真中假定目标初始位置为 $(x_0, y_0) = (45, 60)\mathrm{km}$ ，初始速度 $(\dot{x}_0, \dot{y}_0) = (120, 160)\mathrm{m/s}$ ，x、y 轴的过程噪声为 $0.1\mathrm{m/s^2}$， ω 的随机游动误差为 1e-4rad/s， T=2s 。目标航迹分为 5 段：

（1）0s～200s：以初始速度做匀速直线运动。

（2）200s～380s：第一次转弯机动， $\omega = 0.5\mathrm{deg/s}$ 。

（3）380s～540s：第一次转弯后继续做匀速直线运动。

（4）540s～630s：第二次转弯机动， $\omega = 1.0\mathrm{deg/s}$ 。

（5）630s～900s：第二次转弯后继续做匀速直线运动。

采用雷达传感器进行观测，斜距和方位观测精度分别为 10m 和 0.3deg。初始目标位置和速度估计误差分别为 $(35, 30)\mathrm{m}$ 和 $(50, 50)\mathrm{m/s}$ ，跟踪算法采用 EKF 和 UKF，蒙特卡罗仿真次数为 500，所得结果如图 4.7～图 4.10 所示。

仿真过程中发现，采用单模型的 EKF 算法在第一次机动发生时就开始发散。究其原因，是因为在线性化过程中，由于角速度和速度之间存在耦合，使得速度估计的误差对角速度的估值产生了严重影响，从而偏离真实的运动模型，误差的积累最终导致滤波器发散。相反，由于 UKF 算法在滤波过程中不涉及函数展开，比 EKF 具有更高的精度，因此能对目标航迹进行有效跟踪。

图 4.7 为单次仿真的航迹跟踪图，图 4.8 为跟踪过程中的转弯率估计结果。由图 4.7 和图 4.8 中可以看出，采用单模型 JEUKF 可以实现对转弯目标运动轨迹的准确跟踪且跟踪过程较为稳定，这是因为跟踪过程中实现了对转弯率的有效估计。从图 4.8 的结果看，尽管机动参数的大小估计较为准确，但在目标运动发生模式变化时，转弯率的估计存在一定的滞后现象，这是因为本实例中给出的转弯率是阶跃型的，建立稳定的估计需要一定的时间积累造成的。

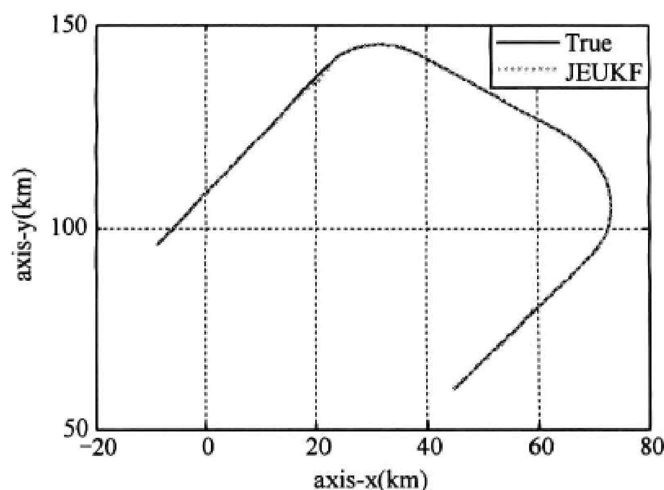


图 4.7　目标航迹单次跟踪

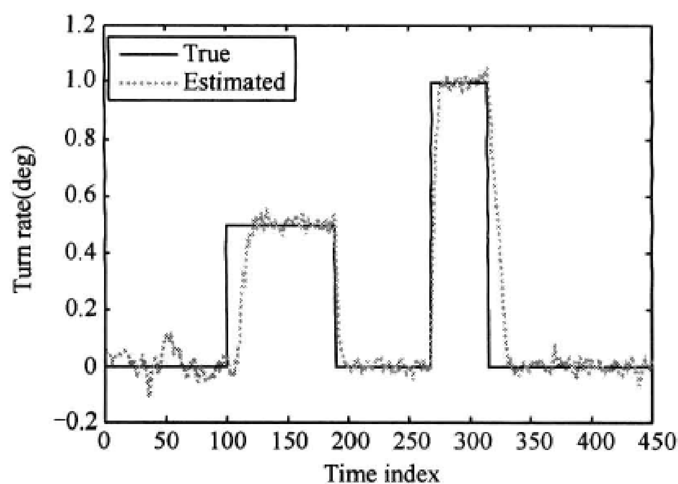


图 4.8　目标转弯率估计

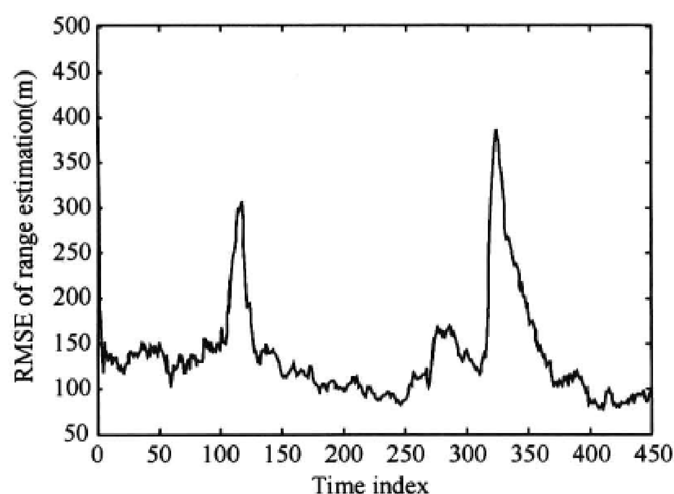


图 4.9　目标距离跟踪均方根误差

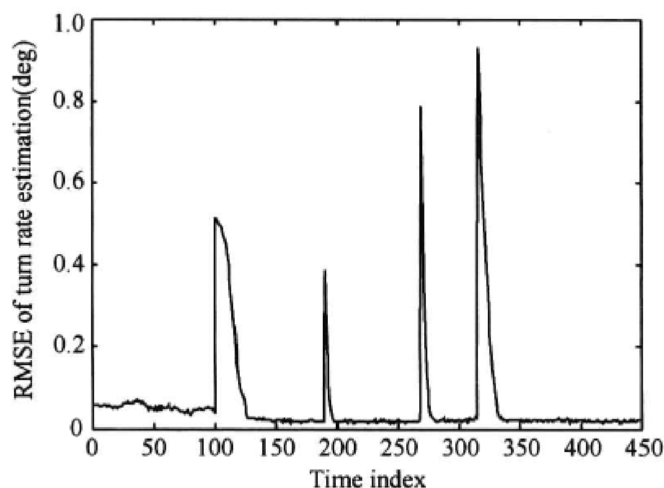


图 4.10　转弯率估计均方根误差

图 4.9 和 4.10 给出的是 500 次实验得到目标距离和转弯率估计均方根误差，由此可以看出，采用联合估计的方法，滤波器能对目标的联合状态进行准确估计，进而对运动模式变化（对应机动的开始或结束）作出快速的响应。机动完成后，转弯率的估值也迅速降至 0 附近，CT 模型因此退化为 CV 模型，这正是采用单模型的 UKF 算法实现机动目标跟踪的原因。与传统的具有 M 个模型的 IMM 算法相比，单模型跟踪算法简单、高效，其运算复杂度低于 IMM 算法的 $1/M$。

4.7　本 章 小 结

从贝叶斯估计的观点来看，滤波问题的关键是如何在给定观测量的基础上建立状态的后验分布，从而求得状态的均值和方差。对于一般的非线性方程，很难得到后验状态分布的解析表达式，为方便处理，通常将这一分布近似为高斯分布。很明显，EKF 和 UKF 都是建立在解析高斯近似的基础上，但是这两类算法的实现途径不同，EKF 及其衍生算法通过泰勒展开将非线性方程进行线性化；而 UKF 则利用 UT 方法对样点进行变换。由于基于 UT 的方法精度高且方便实现，因此 UKF 比传统 EKF 具有很大的优势。作为对比，本章首先详细讨论了 UKF 算法原理和实现方法，在此基础上结合一种新的迭

代策略提出了 UKF 的迭代算法；针对一些特殊的系统，为了降低算法运算开销，提高实时性，介绍了 UKF 的简化算法——SUKF；考虑到含参量的系统估计问题，还研究了用于状态—参数联合估计的 JEUKF 算法。本章最后通过多个实例对前文中的算法进行了仿真测试和实测数据验证。

从跟踪应用的实例结果看，UKF 算法在性能方面要优于 EKF 算法，尤其是在初始误差较大的情况下更是如此。这是因为，当初始误差较大时，由一阶线性化近似导致的误差可能显著增大，严重影响滤波器的性能。另外，迭代型算法在收敛速度方面要优于非迭代型算法，这是因为在滤波过程中进行了多次修正的结果。作为一种较新的非线性滤波算法，UKF 因其近似精度不受非线性程度的影响，且计算较为方便等特点，在实际应用中受到越来越多的关注。

第 5 章　求积近似高斯滤波

通过第 3 章和第 4 章的介绍不难看出，函数近似高斯滤波和确定采样近似高斯滤波都是最优贝叶斯滤波在非线性条件下的两种次优近似，两者既有区别又有联系。前者采用的是直接对非线性函数本身进行近似，而后者则是利用有限个确定的采样点对系统的分布进行近似，但两者都将状态量视为 GRV，以此为前提得到了卡尔曼滤波框架下的各种估计算法。

事实上，从第 1 章贝叶斯滤波的通用框架中可以清楚地看出，滤波过程中涉及的两个重要步骤，即预测和更新，包含了两个积分运算。在高斯假设条件下（即预测分布和后验分布都近似为高斯分布），该积分式等效为高斯函数与非线性函数乘积的多维积分运算，这一特点给近似高斯滤波的实现提供了一个新的思路，各种求积近似滤波方法就是通过该积分式的数值近似来实现的。

本章主要阐述基于求积近似的高斯滤波原理和应用。首先给出了一般非线性方程的高斯—厄米特求积计算原理，由此得到了基于这一计算原理的求积卡尔曼滤波（QKF）算法；考虑到传统求积计算中求积点个数随状态维数的增加呈指数增长关系可能导致的维数灾难问题，本章引入了一种运算相对简单的求容积规则，并得到了求容积卡尔曼滤波（CKF）算法及其求根实现形式；最后给出了目标跟踪的应用实例。

5.1　GH 卡尔曼滤波

5.1.1　GH 求积分规则

考虑下面的单变量函数求积分问题

$$I(g)=\int w(x)g(x)\mathrm{d}x$$

式中：$w(\boldsymbol{x})$ 为权值函数，满足非负性，取 m 个积分点，可近似得到

$$I(g)\approx\sum_{j=1}^{m}w_j g(\chi_j)$$

式中：χ_j 为第 j 个求积分点；w_j 为其权值。

假如给定 m 个积分点，权值的计算可首先通过计算其 k 阶矩 M_k

$$M_i=\int_a^b x^i w(x)\mathrm{d}x\ ,\quad i=0,1,\cdots,m-1$$

进而求解下式所示的矩方程得到

$$
\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \chi_1 & \chi_2 & \cdots & \chi_m \\ \vdots & \vdots & \ddots & \vdots \\ \chi_1^{m-1} & \chi_2^{m-1} & \cdots & \chi_m^{m-1} \end{bmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{pmatrix} = \begin{pmatrix} M_0 \\ M_1 \\ \vdots \\ M_{m-1} \end{pmatrix}
$$

若采用 Gauss-Hermite 求积规则，则权值函数取为标准正态分布，假设 $w(x)=\mathcal{N}(x;0,1)$，此时有

$$
E[g(x)] = \int g(x)\mathcal{N}(x;0,1)\mathrm{d}x \approx \sum_{j=1}^{m} w_j g(\chi_j)
$$

式中，积分点 χ_j 的一种选取方法为：构造一个对角对称矩阵 $\boldsymbol{J}$，对角上的元素为 0，$\boldsymbol{J}_{i,i+1}=\sqrt{i/2}(i=1,2,\cdots,m-1)$；积分点 $\chi_j=\sqrt{2}\lambda_j$，λ_j 为矩阵 $\boldsymbol{J}$ 的第 j 个特征值，权值 $w_j=[(\boldsymbol{v}_j)_1]^2$，$(\boldsymbol{v}_j)_1$ 为其第 j 个归一化特征向量的第一个元素。

以最常用的 m=3 点求积公式为例，根据上述规则构造矩阵

$$
\boldsymbol{J} = \begin{bmatrix} 0 & \sqrt{2}/2 & 0 \\ \sqrt{2}/2 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}
$$

由此得到的特征值为 $\lambda=0,-\sqrt{3/2},\sqrt{3/2}$，对应的归一化向量分别为

$$
\begin{aligned}
\boldsymbol{v}_1 &= [-\sqrt{2/3}, 0, 1/\sqrt{3}]^{\mathrm{T}} \\
\boldsymbol{v}_2 &= [-1/\sqrt{6}, \sqrt{2}/2, -1/\sqrt{3}]^{\mathrm{T}} \\
\boldsymbol{v}_3 &= [1/\sqrt{6}, \sqrt{2}/2, 1/\sqrt{3}]^{\mathrm{T}}
\end{aligned}
$$

最终可得 3 个积分点为 $\chi=0,\pm\sqrt{3}$，相应地，得 $w=2/3,1/6,1/6$。

对于 n_x 维矢量 $\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{x};\boldsymbol{0},\boldsymbol{I}_{n_x})$，其高斯-厄米特求积规则如下

$$
\begin{aligned}
E[g(\boldsymbol{x})] &= \int g(\boldsymbol{x})\mathcal{N}(\boldsymbol{x};\boldsymbol{0},\boldsymbol{I}_{n_x})\mathrm{d}\boldsymbol{x} \\
&\approx \sum_{j_{n_x}=1}^{m} w_{j_{n_x}} \cdots \sum_{j_1=1}^{m} w_{j_1} g(\chi_{j_1}) = \sum_{j=1}^{m^{n_x}} w_j g(\chi_j)
\end{aligned} \tag{5-1}
$$

式中：$\boldsymbol{\chi}_j=[\chi_{j_1},\chi_{j_2},\cdots,\chi_{j_{n_x}}]^{\mathrm{T}}$ 和 $w_j=\prod_{p=1}^{n_x} w_{j_p}$ 分别表示第 j 个求积分点和其所对应的权值。

进一步假设 $\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{x};\hat{\boldsymbol{x}},\boldsymbol{P})$，并令 $\boldsymbol{P}=\boldsymbol{S}\boldsymbol{S}^{\mathrm{T}}$，$\boldsymbol{y}=\boldsymbol{S}^{-1}(\boldsymbol{x}-\hat{\boldsymbol{x}})$，得

$$
\begin{aligned}
E[g(\boldsymbol{x})] &= \int g(\boldsymbol{x})\mathcal{N}(\boldsymbol{x};\hat{\boldsymbol{x}},\boldsymbol{P})\mathrm{d}\boldsymbol{x} = \int g(\boldsymbol{S}\boldsymbol{y}+\hat{\boldsymbol{x}})\mathcal{N}(\boldsymbol{y};\boldsymbol{0},\boldsymbol{I}_{n_x})\mathrm{d}\boldsymbol{y} \\
&\approx \sum_{j_{n_x}=1}^{m} w_{j_{n_x}} \cdots \sum_{j_1=1}^{m} w_{j_1} g(\boldsymbol{S}[\chi_{j_1},\chi_{j_2},\cdots,\chi_{j_{n_x}}]^{\mathrm{T}}+\hat{\boldsymbol{x}}) \\
&= \sum_{j=1}^{m^{n_x}} w_j g(\boldsymbol{S}\boldsymbol{\chi}_j+\hat{\boldsymbol{x}})
\end{aligned} \tag{5-2}
$$

5.1.2 GH 求积卡尔曼滤波

重新考虑式（3-13）和式（3-14）所述的非线性系统

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}) + \boldsymbol{w}_k \tag{5-3}$$

$$\boldsymbol{z}_k = h(\boldsymbol{x}_k) + \boldsymbol{v}_k \tag{5-4}$$

式中，各变量的含义与前文相同。

在卡尔曼滤波框架下，引入高斯–厄米特求积分规则计算高斯状态经非线性系统传递后的数值积分解，可得求积分卡尔曼滤波（QKF）算法[27,28]。下面将具体分析 QKF 算法两个重要步骤——时间更新和观测更新的计算过程。

1. 时间更新

假设已知 $k-1$ 时刻的概率分布 $p(\boldsymbol{x}_{k-1} \mid \boldsymbol{z}_{1:k-1}) = \mathcal{N}(\boldsymbol{x}_{k-1}; \hat{\boldsymbol{x}}_{k-1}, \boldsymbol{P}_{k-1})$，且过程噪 $p(\boldsymbol{w}_{k-1}) = \mathcal{N}(\boldsymbol{w}; \boldsymbol{0}, \boldsymbol{Q}_{k-1})$，令 $\boldsymbol{P}_{k-1} = \boldsymbol{S}_{k-1}\boldsymbol{S}_{k-1}^{\mathrm{T}}$。由式（5-2）可知，当采用 m 点求积公式时，$k-1$ 时刻第 j 个（$j = 1, 2, \cdots, m^{n_x}$）求积点可表示为

$$\boldsymbol{\xi}_{j,k-1} = \boldsymbol{S}_{k-1} \cdot \boldsymbol{\chi}_j + \hat{\boldsymbol{x}}_{k-1} \tag{5-5}$$

令 $M = m^{n_x}$，结合式（1-13）和式（1-14）可得 k 时刻状态的预测值 $\hat{\boldsymbol{x}}_{k|k-1}$ 及其协方差 $\boldsymbol{P}_{k|k-1}$ 分别为

$$\hat{\boldsymbol{x}}_{k|k-1} = \int_{R^{n_x}} f(\boldsymbol{x}_k)\mathcal{N}(\boldsymbol{x}_k; \hat{\boldsymbol{x}}_{k-1}, \boldsymbol{P}_{k-1})\mathrm{d}\boldsymbol{x}_{k-1} = \sum_{j=1}^{M} w_j f(\boldsymbol{\xi}_{j,k-1}) \tag{5-6}$$

$$\begin{aligned}
\boldsymbol{P}_{k|k-1} &= \int_{R^{n_x}} [f(\boldsymbol{x}_{k-1}) + \boldsymbol{w}_k - \hat{\boldsymbol{x}}_{k-1|k}][f(\boldsymbol{x}_{k-1}) + \boldsymbol{w}_k - \hat{\boldsymbol{x}}_{k-1|k}]^{\mathrm{T}}\mathcal{N}(\boldsymbol{x}_{k-1}; \hat{\boldsymbol{x}}_{k-1}, \boldsymbol{P}_{k-1})\mathrm{d}\boldsymbol{x}_{k-1} \\
&= \int_{R^{n_x}} f(\boldsymbol{x}_{k-1}) f^{\mathrm{T}}(\boldsymbol{x}_{k-1})\mathcal{N}(\boldsymbol{x}_{k-1}; \hat{\boldsymbol{x}}_{k-1}, \boldsymbol{P}_{k-1})\mathrm{d}\boldsymbol{x}_{k-1} - \hat{\boldsymbol{x}}_{k-1|k}\hat{\boldsymbol{x}}_{k-1|k}^{\mathrm{T}} + \boldsymbol{Q}_{k-1} \\
&= \sum_{j=1}^{M} w_j f(\boldsymbol{\xi}_{j,k-1}) f^{\mathrm{T}}(\boldsymbol{\xi}_{j,k-1}) - \hat{\boldsymbol{x}}_{k-1|k}\hat{\boldsymbol{x}}_{k-1|k}^{\mathrm{T}} + \boldsymbol{Q}_{k-1}
\end{aligned} \tag{5-7}$$

2. 量测更新

假设预测概率分布和观测噪声分布可分别表示为 $p(\boldsymbol{x}_k \mid \boldsymbol{z}_{1:k-1}) = \mathcal{N}(\boldsymbol{x}_k; \hat{\boldsymbol{x}}_{k-1|k}, \boldsymbol{P}_{k|k-1})$ 和 $p(\boldsymbol{v}_{k-1}) = \mathcal{N}(\boldsymbol{v}; \boldsymbol{0}, \boldsymbol{R}_k)$，且 $\boldsymbol{P}_{k|k-1} = \boldsymbol{S}_{k|k-1}\boldsymbol{S}_{k|k-1}^{\mathrm{T}}$，则根据预测分布得到的第 j 个求积分点为

$$\boldsymbol{\xi}_{j,k|k-1} = \boldsymbol{S}_{k|k-1} \cdot \boldsymbol{\chi}_j + \hat{\boldsymbol{x}}_{k|k-1}, \quad j = 1, 2, \cdots, M \tag{5-8}$$

类似地，有

$$\hat{z}_{k|k-1} = \int_{R^{n_x}} h(\boldsymbol{x}_k)\mathcal{N}(\boldsymbol{x}_k; \hat{\boldsymbol{x}}_{k|k-1}, \boldsymbol{P}_{k|k-1})\mathrm{d}\boldsymbol{x}_k = \sum_{j=1}^{M} w_j h(\boldsymbol{\xi}_{j,k|k-1}) \tag{5-9}$$

$$\begin{aligned}
\boldsymbol{P}_{zz,k|k-1} &= \int_{R^{n_x}} h(\boldsymbol{x}_k) h^{\mathrm{T}}(\boldsymbol{x}_k)\mathcal{N}(\boldsymbol{x}_k; \hat{\boldsymbol{x}}_{k|k-1}, \boldsymbol{P}_{k|k-1})\mathrm{d}\boldsymbol{x}_k \\
&= \sum_{j=1}^{M} w_j h(\boldsymbol{\xi}_{j,k|k-1}) h^{\mathrm{T}}(\boldsymbol{\xi}_{j,k|k-1}) - \hat{z}_{k|k-1}\boldsymbol{z}_{k|k-1}^{\mathrm{T}} + \boldsymbol{R}_k
\end{aligned} \tag{5-10}$$

$$\begin{aligned}
\boldsymbol{P}_{xz,k|k-1} &= \int_{R^{n_x}} \boldsymbol{x}_k h(\boldsymbol{x}_k)\mathcal{N}(\boldsymbol{x}_k; \hat{\boldsymbol{x}}_{k|k-1}, \boldsymbol{P}_{k|k-1})\mathrm{d}\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1}\boldsymbol{z}_{k|k-1}^{\mathrm{T}} \\
&= \sum_{j=1}^{M} w_j \boldsymbol{\xi}_{j,k|k-1} h^{\mathrm{T}}(\boldsymbol{\xi}_{j,k|k-1}) - \hat{\boldsymbol{x}}_{k|k-1}\boldsymbol{z}_{k|k-1}^{\mathrm{T}}
\end{aligned} \tag{5-11}$$

由此，结合前文介绍的卡尔曼增益及协方差矩阵计算公式，即可得到 QKF 算法，具

体流程如可概括如下。

Alg（5.1）：Quadrature Kalman Filtering

步骤 1：初始化

给定均值和协方差 $\hat{\boldsymbol{x}}_0 = E[\boldsymbol{x}_0]$，$\boldsymbol{P}_0 = \mathrm{cov}(\boldsymbol{x}_0)$；

计算求积点 $\boldsymbol{\chi}_j\,(j=1,2,\cdots,m)$ 及相应的权值 w_j。

步骤 2：时间更新

（1）滤波协方差矩阵分解，即

$$\boldsymbol{P}_{k-1} = \boldsymbol{S}_{k-1}\boldsymbol{S}_{k-1}^{\mathrm{T}}$$

（2）计算求积点，即

$$\boldsymbol{\xi}_{j,k-1} = \boldsymbol{S}_{k-1}\cdot\boldsymbol{\chi}_j + \hat{\boldsymbol{x}}_{k-1}$$

（3）传播求积分点，即

$$\boldsymbol{\xi}_{j,k|k-1} = f(\boldsymbol{\xi}_{j,k-1})$$

（4）状态预测，即

$$\hat{\boldsymbol{x}}_{k|k-1} = \sum_{j=1}^{M} w_j \boldsymbol{\xi}_{j,k|k-1}$$

（5）预测误差协方差，即

$$\begin{aligned}\boldsymbol{P}_{k|k-1} &= \sum_{j=1}^{M} w_j(\boldsymbol{\xi}_{j,k|k-1} - \hat{\boldsymbol{x}}_{k|k-1})(\boldsymbol{\xi}_{j,k|k-1} - \hat{\boldsymbol{x}}_{k|k-1})^{\mathrm{T}} + \boldsymbol{Q}_{k-1} \\ &= \sum_{j=1}^{M} w_j \boldsymbol{\xi}_{j,k|k-1}\boldsymbol{\xi}_{j,k|k-1}^{\mathrm{T}} - \hat{\boldsymbol{x}}_{k|k-1}\hat{\boldsymbol{x}}_{k|k-1}^{\mathrm{T}} + \boldsymbol{Q}_{k-1}\end{aligned}$$

步骤 3：量测更新

（1）预测协方差矩阵分解，即

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{S}_{k|k-1}\boldsymbol{S}_{k|k-1}^{\mathrm{T}}$$

（2）计算求积分点，即

$$\boldsymbol{\xi}_{j,k|k-1} = \boldsymbol{S}_{k|k-1}\cdot\boldsymbol{\chi}_j + \hat{\boldsymbol{x}}_{k|k-1}$$

（3）求积点传播，即

$$\boldsymbol{\mathcal{E}}_{j,k|k-1} = h(\boldsymbol{\xi}_{j,k|k-1})$$

（4）观测的预测，即

$$\hat{\boldsymbol{z}}_{k|k-1} = \sum_{j=1}^{M} w_j h(\boldsymbol{\xi}_{j,k|k-1})$$

（5）预测误差协方差，即

$$\begin{aligned}\boldsymbol{P}_{zz} &= \sum_{j=1}^{M} w_j(\boldsymbol{\mathcal{E}}_{j,k|k-1} - \hat{\boldsymbol{z}}_{k|k-1})(\boldsymbol{\mathcal{E}}_{j,k|k-1} - \hat{\boldsymbol{z}}_{k|k-1})^{\mathrm{T}} + \boldsymbol{R}_k \\ &= \sum_{j=1}^{M} w_j \boldsymbol{\mathcal{E}}_{j,k|k-1}\boldsymbol{\mathcal{E}}_{j,k|k-1}^{\mathrm{T}} - \hat{\boldsymbol{z}}_{k|k-1}\hat{\boldsymbol{z}}_{k|k-1}^{\mathrm{T}} + \boldsymbol{R}_k\end{aligned}$$

（6）计算互协方差矩阵，即

$$\boldsymbol{P}_{xz}=\sum_{j=1}^{M}w_j(\boldsymbol{\xi}_{j,k|k-1}-\hat{\boldsymbol{x}}_{k|k-1})(\boldsymbol{\mathcal{E}}_{j,k|k-1}-\hat{\boldsymbol{z}}_{k|k-1})^{\mathrm{T}}$$
$$=\sum_{j=1}^{M}w_j\boldsymbol{\xi}_{j,k|k-1}\boldsymbol{\mathcal{E}}_{j,k|k-1}^{\mathrm{T}}-\hat{\boldsymbol{x}}_{k|k-1}\hat{\boldsymbol{z}}_{k|k-1}^{\mathrm{T}}$$

（7）计算卡尔曼增益，即

$$\boldsymbol{K}_k=\boldsymbol{P}_{xz}\boldsymbol{P}_{zz}^{-1}$$

（8）状态更新，即

$$\hat{\boldsymbol{x}}_k=\hat{\boldsymbol{x}}_{k|k-1}+\boldsymbol{K}_k(\boldsymbol{z}_k-\hat{\boldsymbol{z}}_{k|k-1})$$

（9）状态协方差矩阵更新，即

$$\boldsymbol{P}_k=\boldsymbol{P}_{k|k-1}-\boldsymbol{K}_k\boldsymbol{P}_{zz}\boldsymbol{K}_k^{\mathrm{T}}$$

5.2 求容积卡尔曼滤波

通过 5.1 节的介绍可知，在利用高斯–厄米特求积规则进行积分式近似计算时，其算法复杂度与状态量的维数及在每一维度上求积点的个数密切相关。对于n_x维状态量，若求积点数为m，则根据式（5-1）易知总的求和项为m^{n_x}。求积点的个数决定了近似精度，m个求积点对应$2m-1$阶多项式精度。在m确定的条件下，函数的计算次数随维数n_x呈指数增长，即存在“维数灾难”问题，因此这种方法一般只能适用于维数不太高（如小于 6）的场合。

对于高维函数的求积运算，有必要需求新的解决途径。为此，文献[29]中提出了一种具有线性复杂度且性能优良的求容积方法，并由此导出求容积卡尔曼滤波（CKF）算法，下面对此进行详细介绍。

5.2.1 求容积规则

考虑下面的求积分问题

$$I(f)=\int_{R^{n_x}}f(\boldsymbol{x})\exp(-\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{5-12}$$

首先，将其变换为球径（Spherical-Radial）求容积的形式，然后运用 3 度的球径求容积规则进行求解。具体求解方法如下。

假设$\boldsymbol{x}=r\boldsymbol{y}$，$\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}=1$，得$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}=r^2$，$r\in[0,\infty)$，式（5-12）的积分可在球径坐标系中表示为

$$I(f)=\int_0^{\infty}\int_{U_{n_x}}f(r\boldsymbol{y})r^{n-1}\exp(-r^2)\mathrm{d}(\sigma\boldsymbol{y})\mathrm{d}r \tag{5-13}$$

式中：U_{n_x}表示半径为 1 的球表面，即$U_{n_x}=\{\boldsymbol{y}\in R^{n_x}\mid\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}=1\}$，$\sigma(\cdot)$表示球面测度。

进一步化简，式（5-13）可表示为

$$I(f)=\int_0^{\infty}S(r)r^{n_x-1}\exp(-r^2)\mathrm{d}r \tag{5-14}$$

$$S(r)=\int_{U_{n_x}}f(r\boldsymbol{y})\mathrm{d}\sigma(\boldsymbol{y}) \tag{5-15}$$

对式（5-14）和式（5-15）分别采用m_r点的径（Radial）规则（用高斯–厄米特求积

实现）和 m_s 点的球（Spherical）规则，可得 $m_r \times m_s$ 点的球径求容积规则为

$$\int_0^{\infty} S(r) r^{n_x-1} \exp(-r^2) \mathrm{d}r = \sum_{i=1}^{m_r} a_i S(r_i)$$

$$\int_{U_{n_x}} f(r\boldsymbol{y}) \mathrm{d}\sigma(\boldsymbol{y}) = \sum_{j=1}^{m_s} b_j f(r\boldsymbol{y}_j)$$

$$I(f) = \int_{R^{n_x}} f(\boldsymbol{x}) \exp(-\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \approx \sum_{j=1}^{m_s}\sum_{i=1}^{m_r} a_i b_j f(r_i \boldsymbol{y}_j)$$

取 $m_r = 1$ 和 $m_s = 2n_x$，可得 3 度的球径求容积规则。

根据上述的求容积规则，n_x 维标准正态分布与非线性函数乘积的积分可近似为

$$I(f) = \int_{R^{n_x}} f(\boldsymbol{x}) \mathcal{N}(\boldsymbol{x}; \boldsymbol{0}, \boldsymbol{I}_{n_x}) \mathrm{d}\boldsymbol{x} \approx \sum_{i=1}^{m} w_i f(\boldsymbol{\chi}_i)$$

式中：$m = 2n_x$，$\boldsymbol{\chi}_i = \sqrt{m/2}[\mathbf{1}]_i$；$w_i = 1/m$，$(i = 1, 2, \cdots, m)$；$[\mathbf{1}] \in R^{n_x}$，且

$$[\mathbf{1}]_i \in \left\{ \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}, \cdots, \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \\ \vdots \\ 0 \end{pmatrix}, \cdots, \begin{pmatrix} 0 \\ 0 \\ \vdots \\ -1 \end{pmatrix} \right\}$$

类似于求积规则，假设 $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{x}; \hat{\boldsymbol{x}}, \boldsymbol{P})$，令 $\boldsymbol{P} = \boldsymbol{S}\boldsymbol{S}^{\mathrm{T}}$，得

$$I(f) = \int_{R^{n_x}} f(\boldsymbol{x}) \mathcal{N}(\boldsymbol{x}; \hat{\boldsymbol{x}}, \boldsymbol{P}) \mathrm{d}\boldsymbol{x} \approx \sum_{i=1}^{m} w_i f(\boldsymbol{S}\boldsymbol{\chi}_i + \hat{\boldsymbol{x}})$$

5.2.2 求容积卡尔曼滤波

求容积卡尔曼滤波的形式与求积分卡尔曼滤波相同，仅用求容积点代替其中的求积分点而得到，具体流程可概括如下。

Alg（5.2）：Cubature Kalman Filtering

步骤 1：初始化

给定均值和协方差 $\hat{\boldsymbol{x}}_0 = E[\boldsymbol{x}_0]$，$\boldsymbol{P}_0 = \mathrm{cov}(\boldsymbol{x}_0)$；

计算求容积点 $\boldsymbol{\chi}_j (j=1,2,\cdots,m)$ 及相应的权值 w_j。

步骤 2：时间更新

（1）滤波协方差矩阵分解，即

$$\boldsymbol{P}_{k-1} = \boldsymbol{S}_{k-1}\boldsymbol{S}_{k-1}^{\mathrm{T}}$$

（2）计算求积点，即

$$\boldsymbol{\xi}_{j,k-1} = \boldsymbol{S}_{k-1} \cdot \boldsymbol{\chi}_j + \hat{\boldsymbol{x}}_{k-1}$$

（3）传播求积分点，即

$$\boldsymbol{\xi}_{j,k|k-1} = f(\boldsymbol{\xi}_{j,k-1})$$

（4）状态预测，即

$$\hat{\boldsymbol{x}}_{k|k-1} = \sum_{j=1}^{m} w_j \boldsymbol{\xi}_{j,k|k-1}$$

（5）状态预测协方差矩阵，即

$$\boldsymbol{P}_{k|k-1}=\sum_{j=1}^{m}w_j\boldsymbol{\xi}_{j,k|k-1}\boldsymbol{\xi}_{j,k|k-1}^{\mathrm{T}}-\hat{\boldsymbol{x}}_{k|k-1}\hat{\boldsymbol{x}}_{k|k-1}^{\mathrm{T}}+\boldsymbol{Q}_{k-1}$$

步骤 3：量测更新

（1）矩阵分解，即

$$\boldsymbol{P}_{k|k-1}=\boldsymbol{S}_{k|k-1}\boldsymbol{S}_{k|k-1}^{\mathrm{T}}$$

（2）计算求积分点，即

$$\boldsymbol{\xi}_{j,k|k-1}=\boldsymbol{S}_{k|k-1}\cdot\boldsymbol{\chi}_j+\hat{\boldsymbol{x}}_{k|k-1}$$

（3）求积点传播，即

$$\boldsymbol{\mathcal{E}}_{j,k|k-1}=h(\boldsymbol{\xi}_{j,k|k-1})$$

（4）观测的预测，即

$$\hat{\boldsymbol{z}}_{k|k-1}=\sum_{j=1}^{m}w_j h(\boldsymbol{\xi}_{j,k|k-1})$$

（5）预测误差协方差，即

$$\boldsymbol{P}_{zz}=\sum_{j=1}^{m}w_j(\boldsymbol{\mathcal{E}}_{j,k|k-1}-\hat{\boldsymbol{z}}_{k|k-1})(\boldsymbol{\mathcal{E}}_{j,k|k-1}-\hat{\boldsymbol{z}}_{k|k-1})^{\mathrm{T}}+\boldsymbol{R}_k$$

（6）计算互协方差矩阵，即

$$\boldsymbol{P}_{xz}=\sum_{j=1}^{m}w_j(\boldsymbol{\xi}_{j,k|k-1}-\hat{\boldsymbol{x}}_{k|k-1})(\boldsymbol{\mathcal{E}}_{j,k|k-1}-\hat{\boldsymbol{z}}_{k|k-1})^{\mathrm{T}}$$

（7）计算卡尔曼增益，即

$$\boldsymbol{K}_k=\boldsymbol{P}_{xz}\boldsymbol{P}_{zz}^{-1}$$

（8）状态更新，即

$$\hat{\boldsymbol{x}}_k=\hat{\boldsymbol{x}}_{k|k-1}+\boldsymbol{K}_k(z_k-\hat{\boldsymbol{z}}_{k|k-1})$$

（9）状态协方差矩阵更新，即

$$\boldsymbol{P}_k=\boldsymbol{P}_{k|k-1}-\boldsymbol{K}_k\boldsymbol{P}_{zz}\boldsymbol{K}_k^{\mathrm{T}}$$

5.2.3 平方根实现算法

在工程应用中，CKF 算法在递推计算时，由于计算机字长有限而产生的舍入误差可能导致误差协方差矩阵失去对称性和正定性，从而影响滤波器的数值稳定性。SR-CKF 通过引入正交三角分解回避了直接对矩阵开方运算，可提高滤波器的稳定性。

SR-CKF 中方差阵可写为 $\boldsymbol{P}=\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$，考虑正交三角分解 $\boldsymbol{A}^{\mathrm{T}}=\boldsymbol{Q}\boldsymbol{R}$，$\boldsymbol{Q}$ 为正交阵，$\boldsymbol{R}$ 为上三角阵，于是有

$$\boldsymbol{P}=\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}=\boldsymbol{R}^{\mathrm{T}}\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{R}=\boldsymbol{R}^{\mathrm{T}}\boldsymbol{R}=\boldsymbol{S}\boldsymbol{S}^{\mathrm{T}}$$

引入下列三组记号：

（1）$\boldsymbol{S}=\mathrm{Tria}(\boldsymbol{A})$ 表示一种三角化运算。

（2）$\boldsymbol{S}_{\boldsymbol{Q}}(k)$ 和 $\boldsymbol{S}_{\boldsymbol{R}}$ 分别表示过程噪声和测量噪声方差的平方根。

假设在 k 时刻已经得到 $k-1$ 时刻的滤波估计 $\hat{\boldsymbol{x}}_{k-1}$ 及其误差协方差阵的平方根矩阵

$\boldsymbol{S}_{k-1}$，则k时刻的状态估计值$\hat{\boldsymbol{x}}_k$及其方差阵的平方根$\boldsymbol{S}_k$可以利用 SR-CKF 递推计算求得，具体步骤可概括如下。

Alg（5.3）：Square-Root Cubature Kalman Filtering

1. 时间更新

（1）计算求容积点（j=1,2,⋯,m），即

$$\boldsymbol{\xi}_{j,k-1}=\boldsymbol{S}_{k-1}\cdot\boldsymbol{\chi}_j+\hat{\boldsymbol{x}}_{k-1}$$

式中：m=2n_x（n_x为状态量的维数）；参数$\boldsymbol{\chi}_j$（$\boldsymbol{\varepsilon}_j$为$n_x$阶单位向量）由下式给出

$$\boldsymbol{\chi}_j=\begin{cases}\sqrt{m/2}\cdot\boldsymbol{\varepsilon}_j, & j=1,2,\cdots,n_x\\ -\sqrt{m/2}\cdot\boldsymbol{\varepsilon}_{j-n_x}, & j=n_x+1,n_x+2,\cdots,2n_x\end{cases}$$

（2）计算传播后的求容积点，即

$$\boldsymbol{\xi}_{j,k|k-1}=f(\boldsymbol{\xi}_{j,k-1})$$

（3）状态预测估计值，即

$$\hat{\boldsymbol{x}}_{k|k-1}=\sum_{j=1}^{m}w_j\boldsymbol{\xi}_{j,k|k-1}$$

（4）预测误差方差阵的平方根，即

$$\boldsymbol{S}_{k|k-1}=\text{Tria}([\boldsymbol{\xi}_{k|k-1}\ \ \boldsymbol{S}_{\boldsymbol{Q}}])$$

式中

$$\boldsymbol{\xi}_{k|k-1}=[\boldsymbol{\xi}_{1,k|k-1}-\hat{\boldsymbol{x}}_{k|k-1},\boldsymbol{\xi}_{2,k|k-1}-\hat{\boldsymbol{x}}_{k|k-1},\cdots,\boldsymbol{\xi}_{m,k|k-1}-\hat{\boldsymbol{x}}_{k|k-1}]\boldsymbol{W}$$

$$\boldsymbol{W}=\text{diag}[\sqrt{w_1},\sqrt{w_2},\cdots,\sqrt{w_m}]$$

2. 测量更新

（1）计算求容积点，即

$$\boldsymbol{\xi}_{j,k|k-1}=\boldsymbol{S}_{k|k-1}\cdot\boldsymbol{\chi}_j+\hat{\boldsymbol{x}}_{k|k-1}$$

（2）计算传播后的求容积点，即

$$\boldsymbol{\varepsilon}_{j,k|k-1}=h(\boldsymbol{\xi}_{j,k|k-1})$$

（3）测量预测，即

$$\hat{z}_{k|k-1}=\sum_{j=1}^{m}w_j h(\boldsymbol{\xi}_{j,k|k-1})$$

（4）计算方差阵的平方根，即

$$\boldsymbol{S}_{zz,k|k-1}=\text{Tria}([\boldsymbol{\varepsilon}_{k|k-1}\ \ \boldsymbol{S}_{\boldsymbol{R}}])$$

$$\boldsymbol{\varepsilon}_{k|k-1}=[\boldsymbol{\varepsilon}_{1,k|k-1}-\hat{z}_{k|k-1},\boldsymbol{\varepsilon}_{2,k|k-1}-\hat{z}_{k|k-1},\cdots,\boldsymbol{\varepsilon}_{m,k|k-1}-\hat{z}_{k|k-1}]$$

（5）计算互协方差矩阵，即

$$\boldsymbol{P}_{xz}=\sum_{j=1}^{m}w_j(\boldsymbol{\xi}_{j,k|k-1}-\hat{\boldsymbol{x}}_{k|k-1})(\boldsymbol{\varepsilon}_{j,k|k-1}-\hat{z}_{k|k-1})^{\mathrm{T}}=\boldsymbol{\xi}_{k|k-1}\boldsymbol{\varepsilon}_{k|k-1}^{\mathrm{T}}$$

（6）计算卡尔曼增益，即

$$\boldsymbol{K}_k=(\boldsymbol{P}_{xz}/\boldsymbol{S}_{zz}^{\mathrm{T}})/\boldsymbol{S}_{zz}^{\mathrm{T}}$$

（7）状态更新，即

$$\hat{\boldsymbol{x}}_k=\hat{\boldsymbol{x}}_{k|k-1}+\boldsymbol{K}_k(\boldsymbol{z}_k-\hat{\boldsymbol{z}}_{k|k-1})$$

（8）状态协方差矩阵更新，即

$$\boldsymbol{S}_k=\mathrm{Tria}([\hat{\boldsymbol{x}}_{k|k-1}-\boldsymbol{K}_k\hat{\boldsymbol{z}}_{k|k-1}\quad \boldsymbol{K}_k\boldsymbol{S}_{\boldsymbol{R}}])$$

5.3 应用实例

5.3.1 常规点目标跟踪

考虑二维平面中的目标跟踪问题，假定k时刻目标的状态为$\boldsymbol{x}_k=[x\ \dot{x}\ y\ \dot{y}]^{\mathrm{T}}(k)$，系统状态的演化过程满足

$$\boldsymbol{x}_k=\boldsymbol{F}\boldsymbol{x}_{k-1}+\boldsymbol{w}_k \tag{5-16}$$

式中

$$\boldsymbol{F}=\begin{bmatrix}\boldsymbol{F}_1 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{F}_1\end{bmatrix},\quad \boldsymbol{F}_1=\begin{bmatrix}1 & T\\ 0 & 1\end{bmatrix} \tag{5-17}$$

T为采样间隔；$\boldsymbol{w}_k$为零均值高斯过程噪声，其协方差矩阵为

$$\boldsymbol{Q}_k=q^2\cdot\begin{bmatrix}\boldsymbol{Q}_1 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{Q}_1\end{bmatrix},\quad \boldsymbol{Q}_1=\begin{bmatrix}T^3/3 & T^2/2\\ T^2/2 & T\end{bmatrix} \tag{5-18}$$

传感器位于坐标原点，观测方程为

$$\boldsymbol{y}_k=h(\boldsymbol{x}_k)+\boldsymbol{v}_k=\begin{bmatrix}\sqrt{x_k^2+y_k^2}\\ \arctan(y_k/x_k)\end{bmatrix}+\begin{bmatrix}v_{r,k}\\ v_{\varphi,k}\end{bmatrix} \tag{5-19}$$

式中：$\boldsymbol{v}_k$为零均值高斯噪声且$\boldsymbol{R}_k=\mathrm{cov}[\boldsymbol{v}_k]=\mathrm{diag}([\sigma_r^2\ \sigma_\theta^2])$。

仿真中，假定目标初始状态为$\boldsymbol{x}_0=[40\mathrm{km}\ \ -100\sqrt{3}\mathrm{m/s}\ \ 60\mathrm{km}\ \ -100\mathrm{m/s}]^{\mathrm{T}}$，初始估计误差协方差矩阵为$\boldsymbol{P}_0=\mathrm{diag}([300^2,100^2,300^2,100^2])$，$q=2\mathrm{m/s}^2$，$T=1\mathrm{s}$，$\sigma_r=100\mathrm{m}$，$\sigma_\varphi=0.3^\circ$。采用3点高斯–厄米特求积的QKF、CKF及EKF算法分别对目标进行跟踪，实验次数为500，所得结果如图5.1～图5.5所示。作为比较，图中同时给出了目标跟踪的误差下限PCRB。

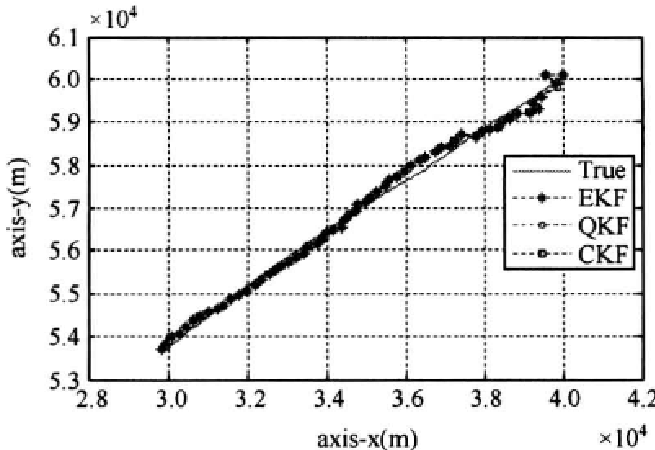


图5.1 单次跟踪示例

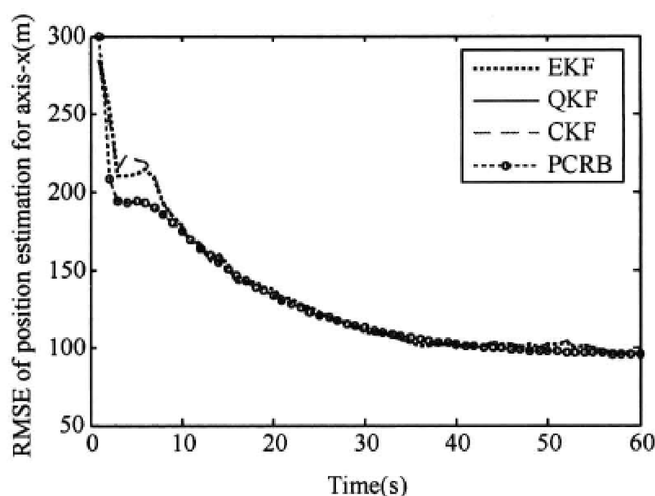


图 5.2　x 轴位置估计均方根误差

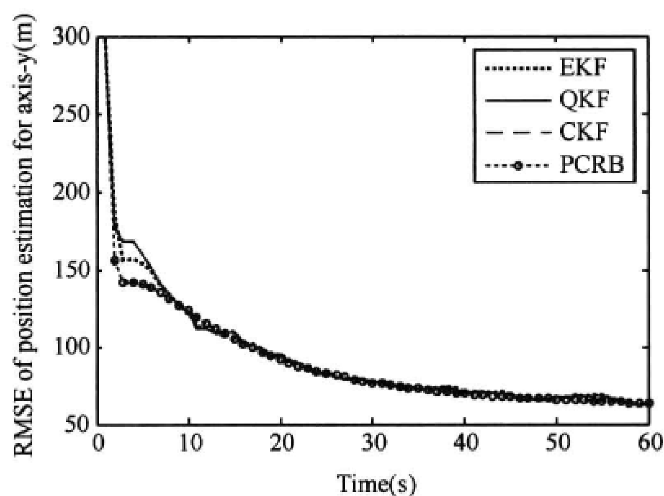


图 5.3　y 轴位置估计均方根误差

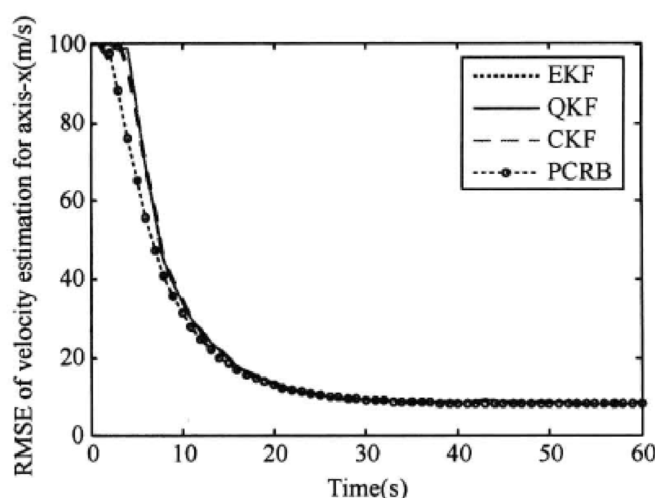


图 5.4　x 轴速度估计均方根误差

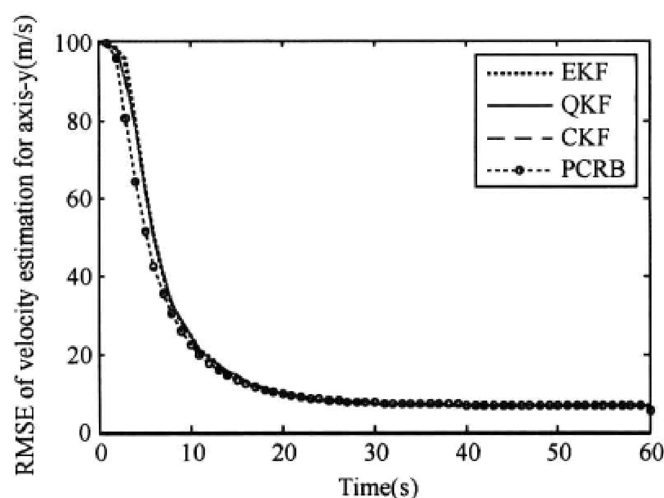


图 5.5　y 轴速度估计均方根误差

由图 5.2～图 5.5 中的结果可以看出，三种算法的估计性能非常接近，尤其是在跟踪收敛之后，几种算法的误差曲线几乎重合。综合本实例及 3.4.1 中的结果可以看出，在系统（这里主要是观测方程）非线性程度不高、且初始误差适中的条件下，EKF 具有与 CDKF、QKF 及 CKF 类似的性能，且从图 5.1～图 5.5 的结果看，这一性能已接近 PCRB 误差曲线，说明达到了近似最优的估计。

表 5.1 给出了三种算法的执行效率比较，由此可以看出，单次仿真中 CKF 的平均时间开销约为 EKF 的 3 倍，而 QKF 的时间消耗比 CKF 稍高，这与 QKF 算法中利用了更多的求积点计算的事实是一致的。

表 5.1　三种算法运行时间比较

算法 / 执行时间	EKF	QKF	CKF
Mean/（s/run）	0.0040	0.0143	0.0125
STD/（s/run）	2.07e-4	4.39e-4	3.44e-4

5.3.2　再入目标跟踪

进一步考虑再入目标跟踪问题，假定 k 时刻目标的状态为 $\boldsymbol{x}_k=[x\ \dot{x}\ y\ \dot{y}]^{\mathrm{T}}(k)$，系统状态的演化过程满足

$$x_k=f(x_{k-1})+w_k=Fx_{k-1}+Gf_d(x_{k-1})+Gf_g+w_k \tag{5-20}$$

式中：$F=\begin{bmatrix}F_1 & 0\\0 & F_1\end{bmatrix}$，$F_1=\begin{bmatrix}1 & T\\0 & 1\end{bmatrix}$；$G=\begin{bmatrix}G_1 & 0\\0 & G_1\end{bmatrix}$，$G_1=\begin{bmatrix}T^2/2\\T\end{bmatrix}$；$w_k$ 为零均值高斯过程噪声，且其协方差矩阵为 $Q_k=q^2\cdot\begin{bmatrix}Q_1 & 0\\0 & Q_1\end{bmatrix}$，$Q_1=\begin{bmatrix}T^3/3 & T^2/2\\T^2/2 & T\end{bmatrix}$；$T$ 为采样间隔；$f_d(x_{k-1})=-0.5\dfrac{g}{\beta}\rho(y_{k-1})\sqrt{\dot{x}_{k-1}^2+\dot{y}_{k-1}^2}[\dot{x}_{k-1}\ \dot{y}_{k-1}]^{\mathrm{T}}$ 为气动阻力对再入目标所产生的阻力加速度；$f_g=[0-\mathrm{g}]^{\mathrm{T}}$ 为地球引力产生的加速度；ρ 为空气密度，当高度小于某一值后，可建模为随目标高度下降按指数衰减的函数，即 $\rho(y)=c_1\mathrm{e}^{-c_2y}$（$c_1$、$c_2$ 为常数）。

假定传感器位于坐标原点，观测方程为

$$y_k=h(x_k)+v_k=\begin{bmatrix}\sqrt{x_k^2+y_k^2}\\\arctan(y_k/x_k)\end{bmatrix}+\begin{bmatrix}v_{r,k}\\v_{\theta,k}\end{bmatrix} \tag{5-21}$$

式中：v_k 为零均值高斯噪声且 $R_k=\mathrm{cov}[v_k]=\mathrm{diag}([\sigma_r^2\ \sigma_\theta^2])$。

假定目标初始状态为 $x_0=[200\mathrm{km}\ \ -1.5\sqrt{3}\mathrm{km/s}\ \ 60\mathrm{km}\ \ -1.5\mathrm{km/s}]^{\mathrm{T}}$，目标状态的初始估计误差协方差矩阵为 $P_0=\mathrm{diag}([1000^2,100^2,1000^2,100^2])$，其他参数设置见表 5.2。

表 5.2　仿真参数设置

q	T	σ_r	σ_θ	c_1	c_2	g	β
$2\mathrm{m/s^2}$	1s	50m	0.3°	1.21907	9146.64	$9.8\mathrm{m/s^2}$	$4\times10^4\mathrm{kg/(m\cdot s^2)}$

由于本实例中的系统方程具有较强的非线性，若采用 EKF 算法求 Jacobian 矩阵将非常繁杂，因此这里主要考虑采用免导数计算的 UKF、QKF 及 CKF 算法分别对目标进行跟踪，仿真次数为 500，所得结果如图 5.6～图 5.10 所示。

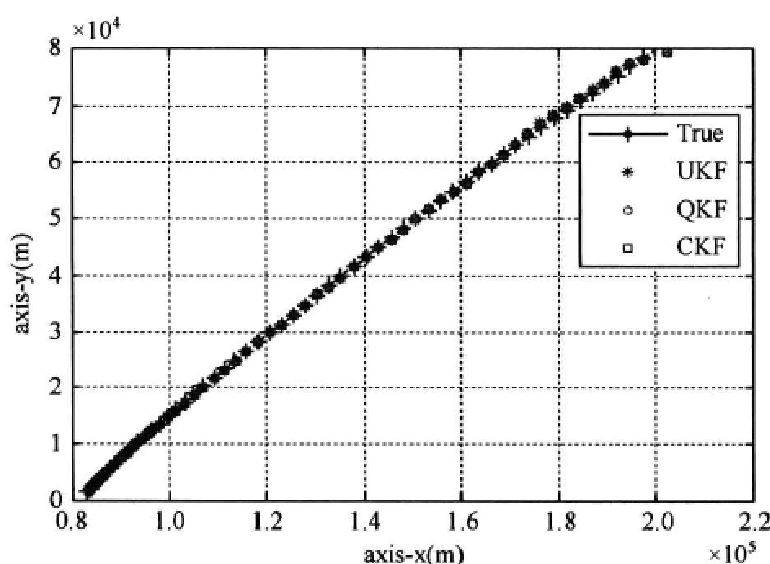


图 5.6　单次跟踪示例

由图 5.6～图 5.10 中的结果看，与前一实例中完全一致的是，QKF 及 CKF 具有完全相似的性能，在图中表现为两种算法的误差曲线几乎完全重合；相比之下，UKF 算法在跟踪达到稳定前的性能略微占点优势。表 5.3 给出了几种算法的时间复杂度比较，由此

可见 CKF 与 QKF 时间开销较为接近，但比 QKF 稍低，这与前面的仿真结果一致。从表中的结果同样可以看出，UKF 算法的时间开销稍大，这主要是源于两方面的原因：一是算法实现时采用的是 UKF 的通用算法，未考虑加性噪声条件下的简化，而后面两种算法直接利用了加性高斯噪声的特点；二是在 QKF 和 CKF 算法中计算预测样点时采用了矩阵乘积运算，UKF 中采用的则是逐点计算的方法。综合比较得知，CKF 在处理此类问题中具有一定的折中优势。

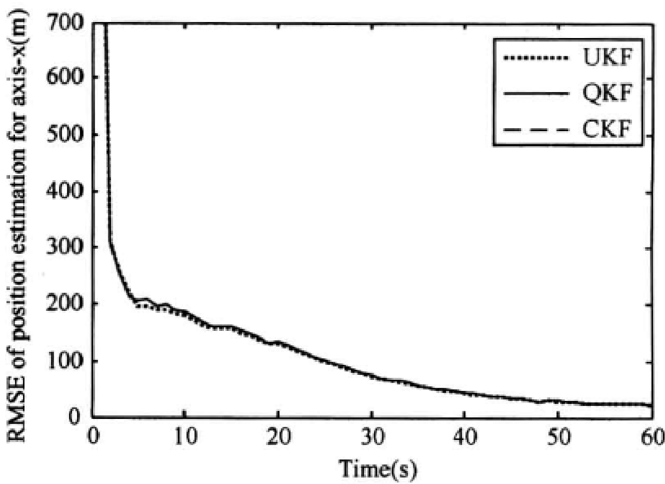


图 5.7 x 方向位置估计均方根误差

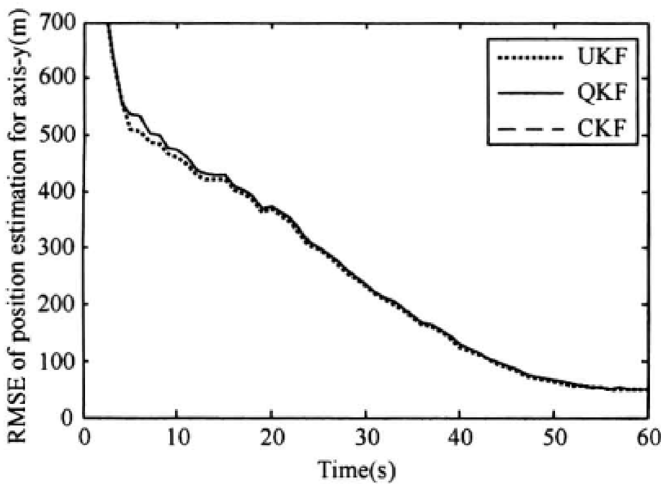


图 5.8 y 方向位置估计均方根误差

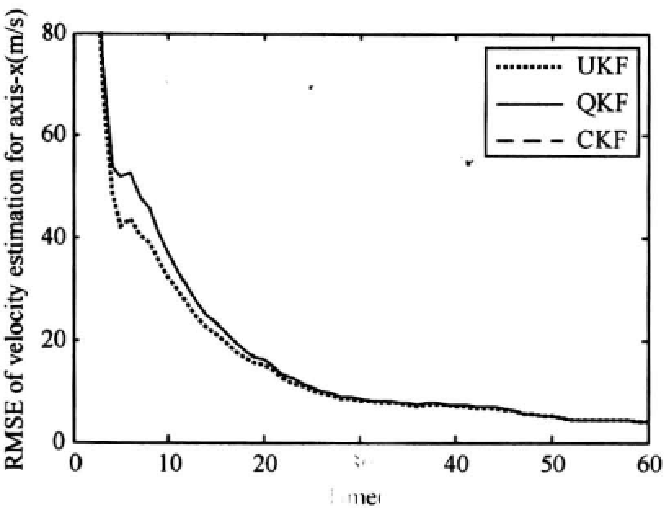


图 5.9 x 方向速度估计均方根误差

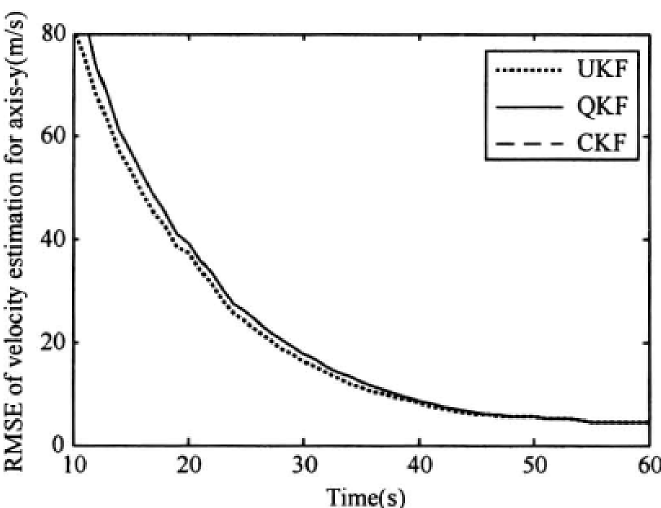


图 5.10 y 方向速度估计均方根误差

表 5.3 三种算法运行时间比较

执行时间 \ 算法	UKF	QKF	CKF
Mean/（s/run）	0.0294	0.0191	0.0159
Std/（s/run）	6.67e-4	5.17e-4	4.40e-4

5.4 本章小结

对于一般的非线性系统，在卡尔曼滤波框架下，状态的预测分布和后验分布都被近似为高斯分布；同时，由经典贝叶斯滤波的两个步骤可知，状态预测和更新算式可表示

成状态分布与非线性系统函数乘积的多维积分形式，这种积分形式的近似解可通过高斯—厄米特求积公式得到（当然也可通过其他求积分规则得到，但高斯-厄米特积分较为常用）。相应地，基于该计算原理的卡尔曼滤波算法称为求积卡尔曼滤波（QKF）算法。对于 m 点高斯-厄米特求积公式，其近似精度为 $2m-1$ 阶多项式，显然这种方法对于单变量非线性问题的求解来说具有精度和效率方面的优势。理论上，特定的权值条件下，三点高斯—厄米特求积结果与不带尺度因子 λ 的 UT 得到的结果是等效的。从这一意义来说，由此得到的 QKF 和 UKF 也应具有相同的性能，这一点已被前文中的实例所证实（图 5.7～图 5.8 中的两种方法在性能方面的细微差异是由于 UKF 算法中的参数设置引起的）。

对于多维状态量，在相同的精度要求下，高斯-厄米特求积计算所需的求积点个数随状态量的维数呈几何级数增长，这对于高维状态问题的求解是极为不利的。最新的研究表明，利用一种特定的求容积规则，可以在保证近似精度损失不大的前提下有效减小求积点数，从而简化运算量。通过典型的目标跟踪实例证实，基于求容积规则的求容积卡尔曼滤波（CKF）算法在性能方面与 QKF 几乎完全相当，具有较好的扩展应用前景。

第 6 章　混合高斯近似滤波

第 3 章～第 5 章主要从非线性系统不同的近似手段这一角度，阐述了次优非线性滤波估计问题。但是应该注意到，讨论这些方法时隐含了一个重要的假设条件，即系统状态是单模态的，也就是说，无论是预测 PDF 还是后验 PDF 都仅用一个高斯分量来近似。理论和实践都证明这种近似方法的有效性仅对单模态问题成立（假设系统状态和噪声都是高斯分布，一个高斯分量对应一个模态）。

然而现实世界中面临的很多问题往往比单模态问题复杂得多，一个典型的例子是系统噪声（过程/观测噪声）不是理想的高斯分布（如 Gamma 分布或其他更复杂的混合分布），此时，建立在高斯噪声假设基础上的各种高斯近似滤波算法由于模型失配等因素的影响难以获得理想的性能。另一个典型例子是系统状态本身具有多模性，如在目标跟踪领域，单模态的状态量很难对多个目标进行充分、有效描述。对于诸如此类的问题，前面介绍的几类近似方法显得无能为力。

解决这一问题的一种可行途径是采用多模近似方法，即采用多个高斯分量组成混合高斯分布来有效近似多模态问题，混合高斯滤波就是基于这一思想提出来的。需要指出的是，由于任何一种复杂的非线性系统都可能用混合高斯分量的形式达到任意精度的近似，从这一点来看，混合高斯滤波算法自然也适用于单模问题。

本章主要介绍基于混合高斯近似的滤波方法，首先给出了非高斯分布的混合高斯近似原理和方法，在此基础上结合 EKF、UKF 和 QKF 三种典型的滤波框架，得到了相应的混合高斯滤波算法，最后给出了闪烁噪声条件和可变数目条件下的目标跟踪应用实例。

6.1　非线性系统的混合高斯近似

在开始阐述具体算法前，首先介绍一个重要引理。

引理：任意 n 维矢量 $\boldsymbol{x}$ 的概率密度函数 $p(\boldsymbol{x})$ 可通过下式

$$p(\boldsymbol{x})=\sum_{i=1}^{I} w_i \mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_i,\boldsymbol{P}_i)\,,\quad \sum_{i=1}^{I} w_i=1 \tag{6-1}$$

进行近似，且只需分量个数 I 足够大，其近似误差可达到任意小。

关于该引理的详细证明可参阅文献[30]，利用这一近似过程，可以处理一些与非高斯现象相关的经验分布（如 ε 混合噪声，它被广泛应用于描述非高斯噪声环境，如跟踪过程中的闪烁噪声、干扰或杂波等）。通过混合高斯学习算法，混合高斯分布可以按任意近似精度逼近非高斯噪声密度，常见的学习算法有期望最大化（EM）算法和 k–均值聚类算法[31]等。因此，对于如下式所示的系统模型

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}) + \boldsymbol{w}_k \tag{6-2}$$

$$\boldsymbol{y}_k = h(\boldsymbol{x}_k) + \boldsymbol{v}_k \tag{6-3}$$

若其噪声为混合高斯加性噪声，则其预测密度和后验密度也可表示为高斯和的形式。具体地，假设加性过程噪声和观测噪声可分别近似为

$$p(\boldsymbol{w}_k) \approx \sum_{i=1}^{I} \alpha_k(i)\mathcal{N}[\boldsymbol{w}_k;\overline{\boldsymbol{w}}_k(i),\boldsymbol{Q}_k(i)] \tag{6-4}$$

$$p(\boldsymbol{v}_k) \approx \sum_{j=1}^{J} \beta_k(j)\mathcal{N}[\boldsymbol{v}_k;\overline{\boldsymbol{v}}_k(j),\boldsymbol{R}_k(j)] \tag{6-5}$$

式中

$$\sum_{i=1}^{I} \alpha_k(j)=1\text{，}\quad \sum_{j=1}^{J} \beta_k(j)=1$$

进一步假设初始时刻的先验密度也为高斯和分布，即

$$p(\boldsymbol{x}_0)=\sum_{i=1}^{I_0} w_0(i)p_i(\boldsymbol{x}_0)=\sum_{i=1}^{I_0} w_0(i)\mathcal{N}[\boldsymbol{x}_0;\overline{\boldsymbol{x}}_0(i),\boldsymbol{P}_0(i)] \tag{6-6}$$

高斯和滤波要求通过 $k-1$ 时刻的后验分布 $p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1})$ 得到 k 时刻的时间更新先验分布

$$p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}) \approx \sum_{r} \alpha_{k|k-1}(r)\mathcal{N}\left(\boldsymbol{x}_{k-1};\hat{\boldsymbol{x}}_{k|k-1}(r),\boldsymbol{P}_{k|k-1}(r)\right) \tag{6-7}$$

及测量更新后验分布

$$p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) \approx \sum_{n=1} \alpha_k(n)\mathcal{N}\left(\boldsymbol{x}_k;\hat{\boldsymbol{x}}_k(n),\boldsymbol{P}_k(n)\right) \tag{6-8}$$

式（6-8）计算过程中，由于状态转移方程的先验分布 $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ 可表示为

$$p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})=\sum_{j=1}^{J} \beta_k(j)\mathcal{N}\left(\boldsymbol{x}_k;f(\boldsymbol{x}_{k-1})+\overline{\boldsymbol{w}}_k(j),\boldsymbol{Q}_k(j)\right) \tag{6-9}$$

于是有

$$\begin{aligned}
p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1})&=\sum_{j=1}^{J}\sum_{i=1}^{I} \alpha_{k-1}(i)\beta_k(j)\int_{R^{n_x}} \mathcal{N}\left(\boldsymbol{x}_{k-1};\hat{\boldsymbol{x}}_{k-1}(i),\boldsymbol{P}_{k-1}(i)\right)\\
&\quad \mathcal{N}\left(\boldsymbol{x}_k;f(\boldsymbol{x}_{k-1})+\overline{\boldsymbol{w}}_k(j),\boldsymbol{Q}_k(j)\right)\mathrm{d}\boldsymbol{x}_{k-1}\\
&\approx \sum_{j=1}^{J}\sum_{i=1}^{I} \alpha_{k|k-1}(i,j)\mathcal{N}\left(\boldsymbol{x}_k;\hat{\boldsymbol{x}}_{k|k-1}(i,j),\boldsymbol{P}_{k|k-1}(i,j)\right)\\
&=\sum_{r=1}^{I\cdot J} \alpha_{k|k-1}(r)\mathcal{N}\left(\boldsymbol{x}_k;\hat{\boldsymbol{x}}_{k|k-1}(r),\boldsymbol{P}_{k|k-1}(r)\right)
\end{aligned} \tag{6-10}$$

式中：$a_{k|k-1}(r)=\alpha_{k-1}(i)\cdot\beta_k(j)$，$\hat{\boldsymbol{x}}_{k|k-1}(r)$ 和 $\boldsymbol{P}_{k|k-1}(r)$ 则可通过非线性高斯滤波器（EKF、QKF 或 UKF 等）的时间更新步骤得到。

同理，观测方程的似然分布可近似为

$$p(\boldsymbol{z}_k|\boldsymbol{x}_k)=\sum_{p=1}^{L} \mu_k(p)\mathcal{N}\left(\boldsymbol{z}_k,h(\boldsymbol{x}_k)+\overline{\boldsymbol{v}}_k(p),\boldsymbol{R}_k(p)\right) \tag{6-11}$$

由此可得

$$
\begin{aligned}
p(\boldsymbol{x}_k|z_{1:k}) &\approx \sum_{p=1}^{L}\sum_{r=1}^{I\cdot J}\mu_k(l)\alpha_{k|k-1}(r)\int_{R^{n_x}}\mathcal{N}\left(\boldsymbol{x}_k,\hat{\boldsymbol{x}}_{k|k-1}(r),\boldsymbol{P}_{k|k-1}(r)\right) \\
&\cdot\mathcal{N}\left(z_k,h(\boldsymbol{x}_k)+\overline{\boldsymbol{v}}_k(p),\boldsymbol{R}_k(p)\right)\mathrm{d}\boldsymbol{x}_k \\
&=\sum_{p=1}^{L}\sum_{r=1}^{I\cdot J}\mu_k(p)\alpha_k(r)\mathcal{N}\left(\boldsymbol{x}_k,\hat{\boldsymbol{x}}_k(r,p),\boldsymbol{P}_k(r,p)\right) \\
&=\sum_{n=1}^{I\cdot J\cdot L}\alpha_k(n)\mathcal{N}\left(\boldsymbol{x}_k,\hat{\boldsymbol{x}}_k(n),\boldsymbol{P}_k(n)\right)
\end{aligned}
\tag{6-12}
$$

式中：$\eta_k(n)=\mathcal{N}\left(\boldsymbol{y}_k,\hat{\boldsymbol{y}}_{k|k-1}(n),\boldsymbol{P}_{yy}(n)\right)$，$\alpha_k(n)=\alpha_{k|k-1}(r)\cdot\eta_k(n)\cdot\mu_k(p)$，$\hat{\boldsymbol{y}}_{k|k-1}(n)$、$\boldsymbol{P}_{yy}(n)$ 为第 n 个分量观测的预测及预测误差协方差矩阵。

在 MMSE 准则下，最终的状态估计及协方差可通过以下定理计算得到。

设 $p(\boldsymbol{x})$ 为包含 n 个高斯分量的归一化混合高斯分布且

$$
p(\boldsymbol{x})=\sum_{i=1}^{n}w_i\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_i,\boldsymbol{P}_i)，\quad \sum_{i=1}^{n}w_i=1
$$

又设 $p_A(\boldsymbol{x})$ 为具有 $\boldsymbol{\mu}$ 均值、$\boldsymbol{\Sigma}$ 协方差的高斯密度且

$$
\boldsymbol{\mu}=\sum_{i=1}^{n}w_i\boldsymbol{\mu}_i
$$

$$
\boldsymbol{\Sigma}=\sum_{i=1}^{n}w_i[\boldsymbol{P}_i+(\boldsymbol{\mu}-\boldsymbol{\mu}_i)(\boldsymbol{\mu}-\boldsymbol{\mu}_i)^{\mathrm{T}}]
$$

则 $p_A(\boldsymbol{x})$ 与 $p(\boldsymbol{x})$ 的前二阶矩（均值和方差）相等。

同时由式（6-12）可以看出，由于过程噪声和观测噪声的非高斯特性，经过一次完整递推后，高斯分量由 I 变为 IJL。显然，在这种条件下若不进行适当的处理，由于高斯分量的不断增长，混合高斯滤波是很难实现的。接下来将针对这一问题，介绍高斯分量的缩减方法。

高斯分量缩减方法通常可分为三类：①决策指导的方法，即高斯分量的删减根据特定的决策规则来进行；②随机删除方法，即随机选择部分分量进行删除；③拟贝叶斯近似方法。在涉及需要动态删减高斯分量的问题中，拟贝叶斯方法最为常用，通常包含以下几种策略。

（1）剪枝。剪枝的含义即舍弃高斯和中那些影响不显著的分量而保留那些影响显著的分量，通常可通过选择部分权值最大的分量、舍弃部分权值最小的分量或者采用门限的方法进行选择。

（2）合并。在这种策略中，用于两高斯分量 i 与 j 之间的相似性可用 Mahalonobis 距离 d_{ij} 来度量，即

$$
d_{ij}^2=\frac{\alpha_i\alpha_j}{\alpha_i+\alpha_j}(\overline{\boldsymbol{x}}_i-\overline{\boldsymbol{x}}_j)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\overline{\boldsymbol{x}}_i-\overline{\boldsymbol{x}}_j)
\tag{6-13}
$$

式中：$\boldsymbol{\Sigma}$ 为高斯和的组合方差；α_i 和 α_j 两分量的权值。

当 d_{ij}^2 小于某设定门限 ε 时，将两个分量进行合并，得到新的高斯分量为

权值：$\alpha_c=\alpha_i+\alpha_j$

均值：$\overline{\boldsymbol{x}}_c=\dfrac{1}{\alpha_i+\alpha_j}(\alpha_i\overline{\boldsymbol{x}}_i+\alpha_j\overline{\boldsymbol{x}}_j)$

方差：$\boldsymbol{\Sigma}_c=\dfrac{1}{\alpha_i+\alpha_j}[\alpha_i\boldsymbol{\Sigma}_i+\alpha_j\boldsymbol{\Sigma}_j+\dfrac{\alpha_i\alpha_j}{\alpha_i+\alpha_j}(\overline{\boldsymbol{x}}_i-\overline{\boldsymbol{x}}_j)(\overline{\boldsymbol{x}}_i-\overline{\boldsymbol{x}}_j)^{\mathrm{T}}]$

6.2 混合高斯非线性滤波

由前面的分析可知，采用混合高斯分布可以近似非高斯噪声。现在问题的关键是，如何得到式（6-10）中的$\mathcal{N}(\boldsymbol{x}_k,\hat{\boldsymbol{x}}_{k|k-1}(r),\boldsymbol{P}_{k|k-1}(r))$及式（6-12）中的$\mathcal{N}(\boldsymbol{x}_k,\hat{\boldsymbol{x}}_k(n),\boldsymbol{P}_k(n))$。在卡尔曼滤波框架下，根据不同的近似手段，如一阶泰勒展开、UT 变换、高斯-厄米特求积，得到相应条件下的高斯和滤波算法 GM-EKF、GM-UKF、GM-QKF 等。下面对这三种最为典型的滤波方法进行详细分析，其他近似手段得到的滤波算法与此类似。

6.2.1 高斯和扩展卡尔曼滤波

在高斯和扩展卡尔曼滤波（GM-EKF）中，状态的预测分布和滤波分布均被近似为高斯和的形式，假设系统状态、过程噪声及观测噪声的高斯分量数分别为 I、J 和 L，则一次完整的递推过程可概括如下。

Alg（6.1）：Gaussian Mixture Extended Kalman Filtering

步骤 1：初始化

给定初始条件 $w_0(i)$，$\hat{\boldsymbol{x}}_0(i)$，$\boldsymbol{P}_0(i)$ $(i=1,2,\cdots,I)$，使

$$p(\boldsymbol{x}_0)=\sum_{i=1}^{I}w_0(i)\mathcal{N}[\boldsymbol{x}_0(i);\hat{\boldsymbol{x}}_0(i),\boldsymbol{P}_0(i)]$$

步骤 2：预测

$$\boldsymbol{F}_{k-1}(i)=\nabla_{\boldsymbol{x}}f(\boldsymbol{x})\big|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{k-1}(i)} \tag{6-14}$$

$$\hat{\boldsymbol{x}}_{k|k-1}(r)=f[\hat{\boldsymbol{x}}_{k-1}(i)]+\overline{\boldsymbol{w}}_k(j)\qquad r=(i-1)J+j \tag{6-15}$$

$$\boldsymbol{P}_{k|k-1}(r)=\boldsymbol{F}_k(i)\boldsymbol{P}_{k-1}(i)\boldsymbol{F}_k^{\mathrm{T}}(i)+\boldsymbol{Q}_k(j) \tag{6-16}$$

$$w_{k|k-1}(r)=w_{k-1}(i)\beta_k(j) \tag{6-17}$$

步骤 3：修正

$$\boldsymbol{H}_k(r)=\nabla_{\boldsymbol{x}}h(\boldsymbol{x})\big|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{k|k-1}(r)} \tag{6-18}$$

$$\hat{\boldsymbol{y}}_{k|k-1}(r,p)=h[\hat{\boldsymbol{x}}_{k|k-1}(r)]+\overline{\boldsymbol{v}}_k(p) \tag{6-19}$$

$$\boldsymbol{K}_k(r,p)=\boldsymbol{P}_{k|k-1}(r)\boldsymbol{H}_k^{\mathrm{T}}(r)[\boldsymbol{H}_k(r)\boldsymbol{P}_{k|k-1}(r)\boldsymbol{H}_k^{\mathrm{T}}(r)+\boldsymbol{R}_k(p)]^{-1} \tag{6-20}$$

$$\hat{\boldsymbol{x}}_k(n)=\hat{\boldsymbol{x}}_{k|k-1}(r)+\boldsymbol{K}_k(r,p)[\boldsymbol{y}_k-\hat{\boldsymbol{y}}_{k|k-1}(r,p)]\qquad n=(r-1)L+p \tag{6-21}$$

$$\boldsymbol{P}_k(n)=\boldsymbol{P}_{k|k-1}(r)-\boldsymbol{K}_k(r,p)\boldsymbol{H}_k(r)\boldsymbol{P}_{k|k-1}(r) \tag{6-22}$$

$$w_k(n)=\frac{w_{k|k-1}(r)\cdot\eta_k(n)\cdot\mu_k(p)}{\sum_{r=1}^{IJ}\sum_{l=1}^{L}w_{k|k-1}(r)\cdot\eta_k(n)\cdot\mu_k(p)} \tag{6-23}$$

$$\eta(n)=\mathcal{N}\left(\boldsymbol{y}_k;\hat{\boldsymbol{y}}_{k|k-1}(r,p),\boldsymbol{H}_k(r)\boldsymbol{P}_{k|k-1}(r)\boldsymbol{H}_k^{\mathrm{T}}(r)+\boldsymbol{R}_k(p)\right) \tag{6-24}$$

步骤 4：估计

$$\hat{\boldsymbol{x}}_k=\sum_{n=1}^{IJL}\tilde{w}_k(n)\hat{\boldsymbol{x}}_k(n) \tag{6-25}$$

$$\boldsymbol{P}_k=\sum_{n=1}^{IJL}\tilde{w}_k(n)\left(\boldsymbol{P}_k(n)+[\hat{\boldsymbol{x}}_k(n)-\hat{\boldsymbol{x}}_k][\hat{\boldsymbol{x}}_k(n)-\hat{\boldsymbol{x}}_k]^{\mathrm{T}}\right) \tag{6-26}$$

步骤 5：剪枝

$$\{\tilde{w}_k(n),\hat{\boldsymbol{x}}_k(n),\boldsymbol{P}_k(n)\}_{n=1}^{IJL}\rightarrow\{\tilde{w}'_k(i),\hat{\boldsymbol{x}}'_k(i),\boldsymbol{P}'_k(i)\}_{i=1}^{I} \tag{6-27}$$

式中：$\tilde{w}_k(n)$ 为 $w_k(n)$ 的归一化结果。

6.2.2 高斯和不敏卡尔曼滤波

将混合高斯分布模型与 UKF 算法相结合，即可得到高斯和不敏卡尔曼滤波（GM-UKF）算法如下。

Alg（6.2）：Gaussian Mixture Unscented Kalman Filtering

步骤 1：初始化

$$p(\boldsymbol{x}_0)=\sum_{i=1}^{I}w_0(i)\mathcal{N}[\boldsymbol{x}_0(i);\hat{\boldsymbol{x}}_0(i),\boldsymbol{P}_0(i)] \tag{6-28}$$

$$\hat{\boldsymbol{x}}_0(i)=E\left[\boldsymbol{x}_0(i)\right] \tag{6-29}$$

$$\boldsymbol{P}_0(i)=E\left[\left(\boldsymbol{x}_0(i)-\hat{\boldsymbol{x}}_0(i)\right)\left(\boldsymbol{x}_0(i)-\hat{\boldsymbol{x}}_0(i)\right)^{\mathrm{T}}\right] \tag{6-30}$$

步骤 2：样点计算

$$\boldsymbol{\chi}_{k-1}(i)=\left[\hat{\boldsymbol{x}}_{k-1}(i)\quad\hat{\boldsymbol{x}}_{k-1}(i)\pm\sqrt{(n_x+\kappa)\boldsymbol{P}_k(i)}\right] \tag{6-31}$$

步骤 3：时间更新

$$\boldsymbol{\chi}_{k|k-1}(i)=f[\boldsymbol{\chi}_{k-1}(i)] \tag{6-32}$$

$$\hat{\boldsymbol{x}}_{k|k-1}(r)=\sum_{l=0}^{2L}w_l^{(m)}\boldsymbol{\chi}_{l,k|k-1}(i)+\bar{\boldsymbol{w}}_k(j) \tag{6-33}$$

$$\begin{aligned}\boldsymbol{P}_{k|k-1}(r)=\boldsymbol{Q}_k(j)+\sum_{l=0}^{2L}w_l^{(c)}\boldsymbol{\chi}_{l,k|k-1}(i)\boldsymbol{\chi}_{l,k|k-1}^{\mathrm{T}}(i)-\\ [\hat{\boldsymbol{x}}_{k|k-1}(r)-\bar{\boldsymbol{w}}_k(j)][\hat{\boldsymbol{x}}_{k|k-1}(r)-\bar{\boldsymbol{w}}_k(j)]^{\mathrm{T}}\end{aligned} \tag{6-34}$$

$$\boldsymbol{\mathcal{Y}}_{k|k-1}(r)=h[\boldsymbol{\chi}_{k|k-1}(r)] \tag{6-35}$$

$$\hat{\boldsymbol{y}}_{k|k-1}(n)=\sum_{l=0}^{2L}w_l^{(m)}\boldsymbol{\mathcal{Y}}_{l,k|k-1}(r)+\bar{\boldsymbol{v}}_k(p) \tag{6-36}$$

步骤 4：测量更新

$$\boldsymbol{P}_{yy}(r,p)=\sum_{l=0}^{2L}w_l^{(c)}\mathcal{Y}_{l,k|k-1}(r)\mathcal{Y}_{l,k|k-1}^{\mathrm{T}}(r)-[\hat{\boldsymbol{y}}_{k|k-1}(n)-\overline{\boldsymbol{v}}_k(p)][\hat{\boldsymbol{y}}_{k|k-1}(n)-\overline{\boldsymbol{v}}_k(p)]^{\mathrm{T}}+\boldsymbol{R}_k(p) \tag{6-37}$$

$$\boldsymbol{P}_{xy}(r,p)=\sum_{l=0}^{2L}w_l^{(c)}\chi_{l,k|k-1}(r)\mathcal{Y}_{l,k|k-1}^{\mathrm{T}}(r)-\hat{\boldsymbol{x}}_{k|k-1}(r)[\hat{\boldsymbol{y}}_{k|k-1}(n)-\overline{\boldsymbol{v}}_k(p)]^{\mathrm{T}} \tag{6-38}$$

$$\boldsymbol{K}_k(r,p)=\boldsymbol{P}_{xy}(r,p)\boldsymbol{P}_{yy}^{-1}(r,p) \tag{6-39}$$

$$\hat{\boldsymbol{x}}_k(n)=\hat{\boldsymbol{x}}_{k|k-1}(r)+\boldsymbol{K}_k(r,p)[\boldsymbol{y}_k-\hat{\boldsymbol{y}}_{k|k-1}(r,p)]\qquad n=(r-1)L+l \tag{6-40}$$

$$\boldsymbol{P}_k(n)=\boldsymbol{P}_{k|k-1}(r)-\boldsymbol{K}_k(r,p)\boldsymbol{P}_{yy}(r,p)\boldsymbol{K}_k^{\mathrm{T}}(r,p) \tag{6-41}$$

$$w_k(n)=\frac{w_{k|k-1}(r)\cdot\eta_k(n)\cdot\mu_k(p)}{\sum_{r=1}^{IJ}\sum_{l=1}^{L}w_{k|k-1}(r)\cdot\eta_k(n)\cdot\mu_k(p)} \tag{6-42}$$

$$\eta(n)=\mathcal{N}\left(\boldsymbol{y}_k;\hat{\boldsymbol{y}}_{k|k-1}(r,p),\boldsymbol{P}_{yy}(r,p)\right) \tag{6-43}$$

步骤 5：估计输出

$$\hat{\boldsymbol{x}}_k=\sum_{n=1}^{IJP}\tilde{w}_k(n)\hat{\boldsymbol{x}}_k(n) \tag{6-44}$$

$$\boldsymbol{P}_k=\sum_{n=1}^{IJP}\tilde{w}_k(n)\left(\boldsymbol{P}_k(n)+[\hat{\boldsymbol{x}}_k(n)-\hat{\boldsymbol{x}}_k][\hat{\boldsymbol{x}}_k(n)-\hat{\boldsymbol{x}}_k]^{\mathrm{T}}\right) \tag{6-45}$$

步骤 6：分量剪枝

$$\{\tilde{w}_k(n),\hat{\boldsymbol{x}}_k(n),\boldsymbol{P}_k(n)\}_{n=1}^{IJL}\rightarrow\{\tilde{w}'_k(i),\hat{\boldsymbol{x}}'_k(i),\boldsymbol{P}'_k(i)\}_{i=1}^{I} \tag{6-46}$$

6.2.3 高斯和求积卡尔曼滤波

同理，将高斯混合分布模型与 QKF 相结合，即可得到高斯和求积卡尔曼滤波（GM-QKF）算法，具体流程可概括如下。

Alg（6.3）：Gaussian Mixture Quadrature Kalman Filtering

步骤 1：时间更新

（1）矩阵分解，即

$$\boldsymbol{P}_{k-1}(r)=\boldsymbol{S}_{k-1}(r)\boldsymbol{S}_{k-1}^{\mathrm{T}}(r),\quad\alpha_{k|k-1}(r)=\alpha_{k-1}(i)\cdot\beta_k(j)\quad(r=(i-1)\cdot I+j)$$

（2）计算求积点，即

$$\xi_{l,k-1}=\boldsymbol{S}_{k-1}(i)\cdot\chi_l+\hat{\boldsymbol{x}}_{k-1}(i),\quad l=1,2,\cdots,m$$

（3）传播求积分点，即

$$\xi_{l,k|k-1}=f(\xi_{l,k-1})$$

（4）状态预测，即

$$\hat{\boldsymbol{x}}_{k|k-1}(r)=\sum_{l=1}^{m}w_l\xi_{l,(k/k-1)}(i)+\overline{\boldsymbol{w}}_k(j)$$

（5）状态预测协方差矩阵，即

$$\boldsymbol{P}_{k|k-1}(r)=\sum_{l=1}^{m}w_l\xi_{l,k|k-1}(i)\xi_{l,k|k-1}^{\mathrm{T}}(i)-[\hat{\boldsymbol{x}}_{k|k-1}(r)-\overline{\boldsymbol{w}}_k(j)][\hat{\boldsymbol{x}}_{k|k-1}(r)-\overline{\boldsymbol{w}}_k(j)]^{\mathrm{T}}+\boldsymbol{Q}_k(j)$$

步骤 2：量测更新

（1）矩阵分解，即

$$\boldsymbol{P}_{k|k-1}(r)=\boldsymbol{S}_{k|k-1}(r)\boldsymbol{S}_{k|k-1}^{\mathrm{T}}(r)$$

（2）计算求积分点，即

$$\xi_{l,k|k-1}(r)=\boldsymbol{S}_{k|k-1}(r)\cdot\chi_l+\hat{\boldsymbol{x}}_{k|k-1}(r)$$

（3）求积点传播，即

$$\boldsymbol{\mathcal{E}}_{l,k|k-1}(r)=h[\xi_{l,k|k-1}(r)]$$

（4）观测预测，即

$$\hat{z}_{k|k-1}(r,p)=\sum_{l=1}^{m}w_l h[\xi_{l,k|k-1}(r)]+\overline{\boldsymbol{v}}_k(p)$$

（5）观测预测协方差矩阵，即

$$\boldsymbol{P}_{zz}(r,p)=\sum_{l=1}^{m}w_l\boldsymbol{\mathcal{E}}_{l,k|k-1}(r)\boldsymbol{\mathcal{E}}_{l,k|k-1}^{\mathrm{T}}(r)-[\hat{z}_{k|k-1}(r)-\overline{\boldsymbol{v}}_k(p)][\hat{z}_{k|k-1}(r)-\overline{\boldsymbol{v}}_k(p)]^{\mathrm{T}}+\boldsymbol{R}_k(p)$$

（6）计算互协方差矩阵，即

$$\boldsymbol{P}_{xz,k|k-1}(r,p)=\sum_{l=1}^{m}w_l\xi_{l,k|k-1}(r)\boldsymbol{\mathcal{E}}_{l,k|k-1}^{\mathrm{T}}(r)-\hat{\boldsymbol{x}}_{k|k-1}(r)[\hat{z}_{k|k-1}(r)-\overline{\boldsymbol{v}}_k(p)]^{\mathrm{T}}$$

（7）计算卡尔曼增益，即

$$\boldsymbol{K}_k(r,p)=\boldsymbol{P}_{xz}(r,p)\boldsymbol{P}_{zz}^{-1}(r,p)$$

（8）状态更新，即

$$\hat{\boldsymbol{x}}_k(n)=\hat{\boldsymbol{x}}_{k|k-1}(r)+\boldsymbol{K}_k(r,p)[z_k-\hat{z}_{k|k-1}(r,p)]$$

（9）状态协方差矩阵更新，即

$$\boldsymbol{P}_k(n)=\boldsymbol{P}_{k|k-1}(r)-\boldsymbol{K}_k(r,p)\boldsymbol{P}_{zz,k|k-1}(r,p)\boldsymbol{K}_k^{\mathrm{T}}(r,p)$$

上述处理流程中，权值更新算式与高斯分量剪枝方法与 Alg（6.1）中的完全相同，在此不再重复。

6.3 应用实例

6.3.1 闪烁噪声下的目标跟踪

仍以二维平面内的目标跟踪为例，假定系统方程由式（5-16）和式（5-19）组成，式中各参数的含义与前文相同，所不同的是距离和方位观测噪声不是理想的高斯噪声，而是非高斯闪烁噪声。对于雷达闪烁噪声的建模，通常可近似为混合高斯分布或高斯与拉普拉斯的混合分布；对于后则，由于拉普拉斯分布一般难以直接处理，因此一般建模成具有多分量的混合高斯分布，并利用 EM 算法得到各高斯分量的参数及权系数。

为考虑问题方便，假定闪烁噪声建模为具有两个不同分量的混合高斯分布，即

$$p(\boldsymbol{v}_k)=(1-\varepsilon)p_1(\boldsymbol{v}_k)+\varepsilon p_2(\boldsymbol{v}_k)$$
$$=(1-\varepsilon)\mathcal{N}(\boldsymbol{v}_k;\boldsymbol{\mu}_1,\boldsymbol{R}_1)+\varepsilon\mathcal{N}(\boldsymbol{v}_k;\boldsymbol{\mu}_2,\boldsymbol{R}_2)$$

式中：ε 为闪烁噪声出现的概率，该值越大说明目标闪烁越频繁。

仿真中，假定目标初始状态为 $\boldsymbol{x}_0=[20\text{km}\ \ -160\text{m/s}\ \ 25\text{km}\ \ -120\text{m/s}]^{\mathrm{T}}$，初始估计误差协方差矩阵为 $\boldsymbol{P}_0=\text{diag}([200^2,100^2,200^2,100^2])$，其余参数见表 6.1。

表 6.1　仿真参数设置

q	T	σ_{r_1}	σ_{φ_1}	σ_{r_2}	σ_{φ_2}	ε	μ_1	μ_2
1m/s^2	1s	20m	0.2°	100m	1°	0.25	**0**	**0**

采用 EKF、GM-EKF 及 GM-UKF 算法分别对目标进行跟踪，其中 EKF 算法采用矩匹配方法，即

$$\boldsymbol{\mu}=E[\boldsymbol{v}_k]=(1-\varepsilon)\boldsymbol{\mu}_1+\varepsilon\boldsymbol{\mu}_2=\boldsymbol{0}$$
$$\boldsymbol{R}=E[(\boldsymbol{v}_k-\boldsymbol{\mu})(\boldsymbol{v}_k-\boldsymbol{\mu})^{\mathrm{T}}]=(1-\varepsilon)\boldsymbol{R}_1+\varepsilon\boldsymbol{R}_2$$

而 GM-EKF 和 GM-UKF 算法中，系统状态采用 16 个高斯分量，每个分量在真值附近采用 $\boldsymbol{P}_0$ 随机生成，各分量的初始权重相等，即

$$p(\boldsymbol{x}_0)=\sum_{i=1}^{16}w_0(i)\mathcal{N}\left(\boldsymbol{x}_0;\boldsymbol{\mu}_0(i),\boldsymbol{P}_0(i)\right)$$

过程噪声采用单分量，其量值与真实的过程噪声一致；而量测噪声建模为 2 个高斯分量，且其均值和协方差参数与真实闪烁噪声相同。在上述设置条件下，每次递推完成后，状态的高斯分量个数由 16 变为 $16\times2=32$ 个，为保证算法的可实现性，采用取大裁剪法将高斯分量数目恢复为 16。独立运行 500 次蒙特卡罗仿真，所得结果如图 6.1～图 6.5 所示，为图示清楚，图 6.1 给出的是局部放大的结果。

由图中的结果可以，在本仿真参数设置条件下，GM-EKF 和 GM-UKF 的性能非常接近，且比 EKF 算法要有明显改善（特别是对于目标位置估计）。这就说明：①EKF 和 UKF 估计算法本身的性能在该实例中是相当的；②在非高斯（闪烁）噪声条件下，影响目标跟踪精度的主要因素是模型的非高斯性而不是方程的非线性；③传统的基于单高斯分量的卡尔曼型滤波算法在处理非高斯问题时将会导致明显的性能下降。

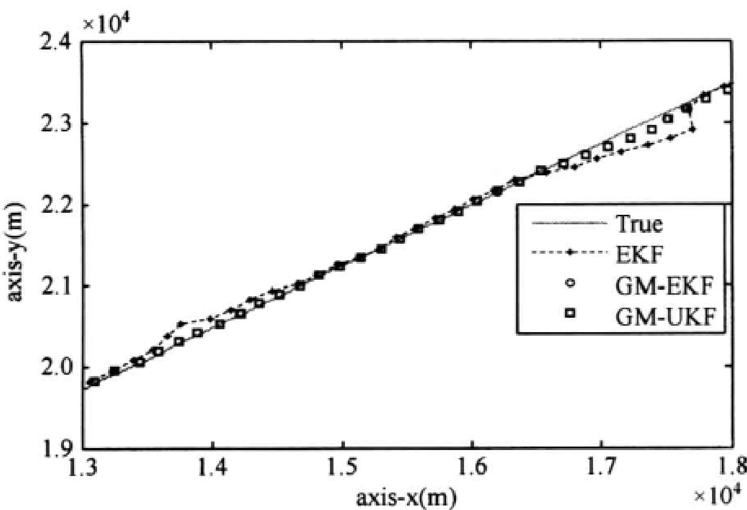


图 6.1　单次跟踪示例

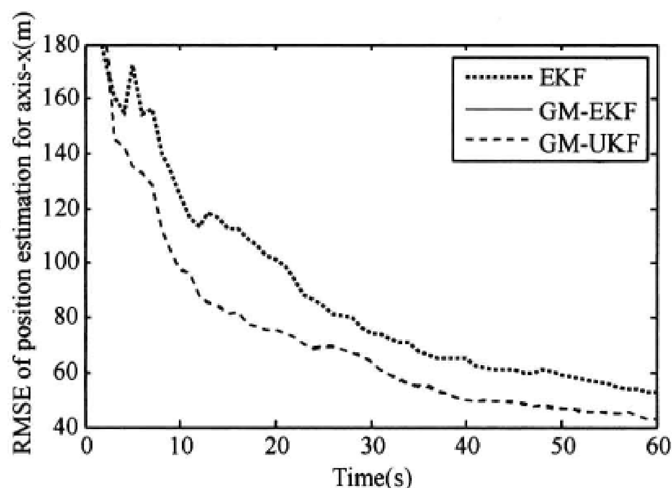


图 6.2 x 轴位置估计均方根误差

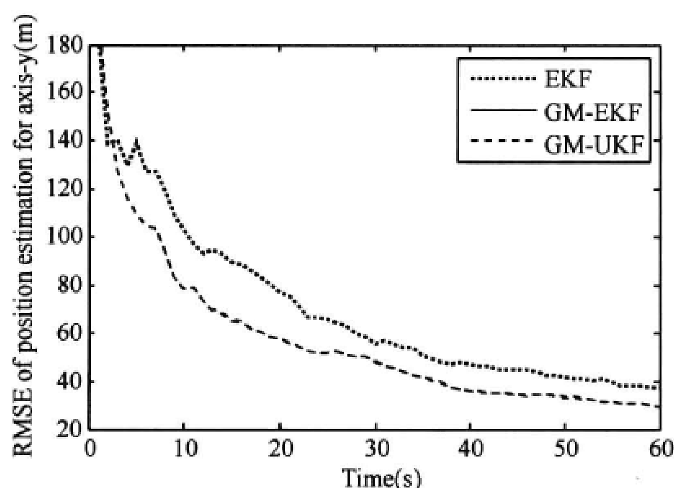


图 6.3 y 轴位置估计均方根误差

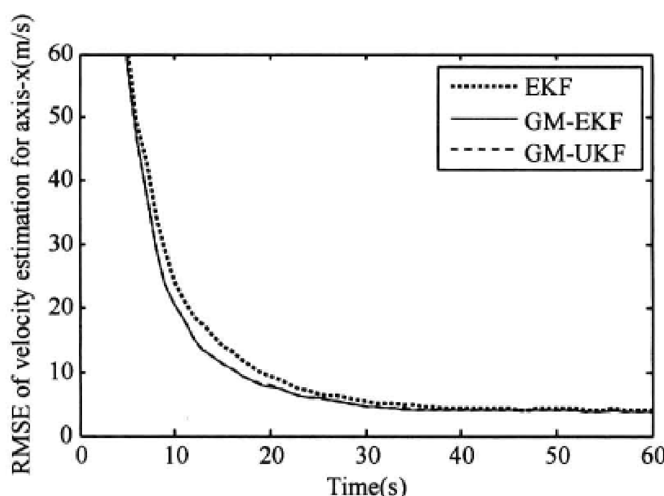


图 6.4 x 轴速度估计均方根误差

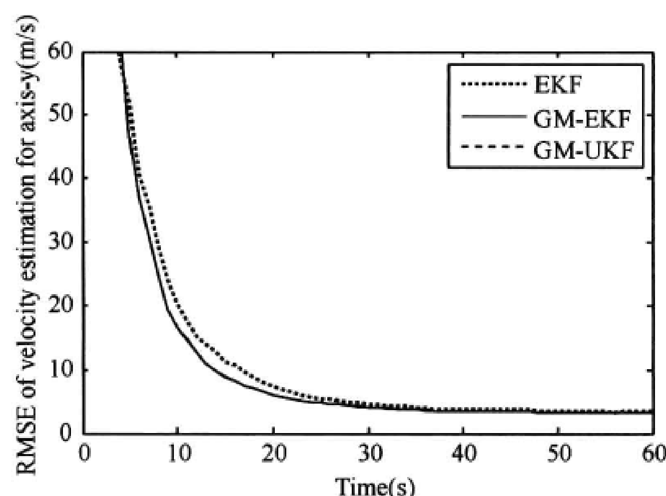


图 6.5 y 轴速度估计均方根误差

为了比较算法的复杂度，表 6.2 给出了三种算法运算时间开销。由此可见，采用混合高斯模型之后，算法的运算开销也相应增大，且其增量略高于高斯分量总数所对应的倍数。不过，由于这种滤波算法具有较好的并行性，若采用并行处理实现手段，在时间开销方面可做到与单高斯分量相当。

表 6.2 三种算法运行时间比较

算法 / 执行时间	EKF	GM-EKF	GM-UKF
Mean/（s/run）	0.0040	0.1765	0.8231
STD/（s/run）	1.46e-4	0.0011	0.0073

为了进一步分析噪声闪烁强度对跟踪性能的影响，表 6.3 给出了跟踪稳定建立后（大于 10s）不同闪烁概率条件下的平均均方根误差（ARMSE）。由表中的结果可以清楚看出，随着闪烁强度的降低，三种跟踪算法的跟踪精度一致提高；但即便是在 ε=0.05 的情况下，两种混合高斯滤波器（GM-EKF 和 GM-UKF）的估计性能仍要明显好于 EKF，说明了混合高斯模型在处理非高斯问题中的有效性。

表 6.3　三种算法平均均方根误差（ARMSE）比较

算法 ＼ 估计误差	pos_x（m）	pos_y（m）	vel_x（m/s）	vel_y（m/s）
EKF ε=0.25	75.9842	57.5557	6.8366	5.6792
ε=0.15	62.6276	47.0547	6.2338	5.0482
ε=0.05	45.7380	34.5908	5.0113	4.2256
GM-EKF ε=0.25	59.8275	44.8822	5.8776	4.8713
ε=0.15	43.4483	32.4868	4.7965	3.9421
ε=0.05	35.0325	26.5295	4.2463	3.6285
GM-UKF ε=0.25	59.9895	45.0083	5.8893	4.8774
ε=0.15	43.3378	32.4104	4.7764	3.9269
ε=0.05	34.9981	26.4758	4.2433	3.6212

6.3.2　GM-PHD 多目标跟踪

本实例主要利用混合高斯近似方法结合概率假设密度（PHD）滤波来实现对多目标的跟踪，为此首先对有限集统计的相关理论作一简要介绍。

有限集统计（Finite Set Statistics，FISST）理论[32]给出了随机有限集（RFS）的统计特性，利用随机集理论处理单/多传感器多目标跟踪问题是行之有效的方法。在 FISST 中，目标集视为一个全局目标，传感器（组）在相同时刻提供的量测认为是一个全局量测，多目标的状态模型和观测模型可以表示为 RFS 形式。

假设在 k 时刻，多目标状态集为 $\boldsymbol{X}_k=\{\boldsymbol{x}_{k,1},\boldsymbol{x}_{k,2},\cdots,\boldsymbol{x}_{k,M_k}\}$，多目标观测集为 $\boldsymbol{Z}_k=\{z_{k,1},z_{k,2},\cdots,z_{k,N_k}\}$。其中，$M_k$ 和 N_k 分别表示 k 时刻的目标数目和量测数目。状态集和观测集分别用状态空间的 RFS $\boldsymbol{X}_k$ 和观测空间的 RFS $\boldsymbol{Z}_k$ 表示，因此，k 时刻的目标状态可用 RFS 描述为

$$\boldsymbol{X}_k=S_{k|k-1}(\boldsymbol{X}_{k-1})\cup B_{k|k-1}(\boldsymbol{X}_{k-1})\cup\Gamma_k \tag{6-47}$$

式中：$\boldsymbol{X}_{k-1}$ 和 $\boldsymbol{X}_k$ 分别为 $k-1$ 和 k 时刻的目标状态 RFS；$S_{k|k-1}$ 表示 k 时刻仍存在的目标 RFS；$B_{k|k-1}$ 为由 $\boldsymbol{X}_{k-1}$ 衍生（Spawn Birth）的目标 RFS；Γ_k 为 k 时刻瞬间出现的新生（New Birth）目标 RFS。

k 时刻的多目标观测模型可用 RFS 描述为

$$\boldsymbol{Z}_k=\Theta_k(\boldsymbol{X}_k)\cup C_k(\boldsymbol{X}_k) \tag{6-48}$$

式中：Θ_k 为 $\boldsymbol{X}_k$ 的观测 RFS；C_k 为由杂波或者虚警引起的观测 RFS。

根据贝叶斯估计理论，基于 RFS 的贝叶斯估计可描述为

$$p_{k|k-1}(\boldsymbol{X}_k|\boldsymbol{Z}_{1:k-1})=\int f_{k|k-1}(\boldsymbol{X}_k|\boldsymbol{X}_{k-1})p_{k-1}(\boldsymbol{X}_{k-1}|\boldsymbol{Z}_{1:k-1})\mu(\mathrm{d}\boldsymbol{X}_{k-1}) \tag{6-49}$$

$$p_k(\boldsymbol{X}_k|\boldsymbol{Z}_{1:k})=\frac{g_k(\boldsymbol{Z}_k|\boldsymbol{X}_k)p_{k|k-1}(\boldsymbol{X}_k|\boldsymbol{Z}_{1:k-1})}{\int g_k(\boldsymbol{Z}_k|\boldsymbol{X}_k)p_{k|k-1}(\boldsymbol{X}_k|\boldsymbol{Z}_{1:k-1})\mu(\mathrm{d}\boldsymbol{X}_k)} \tag{6-50}$$

式中：$p_{k|k-1}$、p_k、$f_{k|k-1}$ 和 g_k 分别为多目标先验概率、后验概率、转移概率和似然函数；μ 为某一空间适当的参考测度。

若能通过贝叶斯理论递推得到系统状态的后验概率密度，就可方便地获取其均值、方差和峰值等估计。但是，式（6-49）和式（6-50）需要进行高维积分运算，并且多目标联合似然函数 g_k 的计算随着目标数的增加而呈指数增长，使得直接用贝叶斯公式来估计目标数目和目标状态变得难以实现。为了简化计算，文献[33]利用多目标后验分布的概率假设密度（PHD）来近似代替多目标的后验概率密度，并以此来递推估算目标数和目标状态，大大减小了计算负担；而文献[34]～文献[36]则在此基础上给出了 PHD 滤波的高斯混合近似实现形式。

概率假设密度 $D_{\Xi}(\boldsymbol{x})$ 表示随机有限集 Ξ 的一阶矩，定义为

$$D_{\Xi}(\boldsymbol{x})=E\left[\sum_{y\in\Xi}\delta_y(\boldsymbol{x})\right]=\int\sum_{y\in\Xi}\delta_y(\boldsymbol{x})f_{\Xi}(\boldsymbol{X})\delta\boldsymbol{X} \tag{6-51}$$

在多目标跟踪中，$D_{\Xi}(\boldsymbol{x}_0)$ 表示在 $\boldsymbol{x}_0$ 点期望的目标密度；$\int D_{\Xi}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$ 表示出现在监视区域内的期望目标数；而 $D_{\Xi}(\boldsymbol{x})$ 的极值，给出了目标的状态估计。

PHD 滤波通过 PHD 的预测和更新方程来实现递推传播，其预测 PHD 可表示为

$$D_{k|k-1}(\boldsymbol{x}_k|\boldsymbol{Z}_{1:k-1})=\gamma_k(\boldsymbol{x}_k)+\int D_{k|k-1}(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})D_{k-1}(\boldsymbol{x}_{k-1}|\boldsymbol{Z}_{1:k-1})\mathrm{d}\boldsymbol{x}_{k-1} \tag{6-52}$$

式中：$D_{k|k-1}(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})=b_{k|k-1}(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})+s_{k|k-1}f(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$；$\gamma_k(\boldsymbol{x}_k)$ 为新生（New）目标的 PHD；$b_{k|k-1}$ 为由存在目标衍生（Spawn）的目标 PHD；$s_{k|k-1}$ 为存活（Survival）概率；$f(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ 为转移概率密度。

当 k 时刻的量测 $\boldsymbol{z}_k$ 可用时，其更新 PHD 为

$$D_k(\boldsymbol{x}_k|\boldsymbol{Z}_{1:k})=(1-P_D)D_{k|k-1}(\boldsymbol{x}_k|\boldsymbol{Z}_{1:k-1})+\sum_{\boldsymbol{z}\in\boldsymbol{Z}_k}\frac{P_D D_k(\boldsymbol{z})}{\lambda_c c_k(\boldsymbol{z})+P_D D_k(\boldsymbol{z})}D_k(\boldsymbol{x}_k|\boldsymbol{z}_k) \tag{6-53}$$

式中：$D_k(\boldsymbol{z})=\int g_k(\boldsymbol{z}|\boldsymbol{x}_k)D_{k|k-1}(\boldsymbol{x}_k|\boldsymbol{Z}_{1:k-1})\mathrm{d}\boldsymbol{x}_k$；$D_k(\boldsymbol{x}_k|\boldsymbol{z}_k)=g_k(\boldsymbol{z}|\boldsymbol{x}_k)D_{k|k-1}(\boldsymbol{x}_k|\boldsymbol{Z}_{1:k-1})/D_k(\boldsymbol{z})$；$g_k(\boldsymbol{z}|\boldsymbol{x}_k)$ 为传感器似然函数；λ_c 为每次扫描的平均虚警个数（通常建模为服从泊松分布）；$c_k(\boldsymbol{z})$ 为每个虚警的分布函数；P_D 为检测概率。

用混合高斯分量来近似多目标状态的分布，并结合 PHD 滤波方法即可得到 GM-PHD 多目标跟踪算法，下面以典型非线性系统方程

$$\boldsymbol{x}_k=f(\boldsymbol{x}_{k-1})+\boldsymbol{w}_k \tag{6-54}$$

$$\boldsymbol{z}_k=h(\boldsymbol{x}_k)+\boldsymbol{v}_k \tag{6-55}$$

的形式来介绍 GM-PHD 算法的具体实现过程。

假设在 k=0 时刻，D_k 初始化为具有 J_k 个分量的混合高斯分布，即

$$D_k(\boldsymbol{x}|\boldsymbol{z}_k)=\sum_{j=1}^{J_k}w_k[j]\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_k[j],\boldsymbol{\Sigma}_k[j]) \tag{6-56}$$

则在预测阶段，新生目标密度和衍生目标可分别用高斯混合形式表示为

$$\gamma_k(\boldsymbol{x})=\sum_{i=1}^{J_{\gamma,k}}w_{\gamma k}[i]\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_{\gamma k}[i],\boldsymbol{\Sigma}_{\gamma k}[i]) \tag{6-57}$$

$$b_{k|k-1}(\boldsymbol{x})=\sum_{j=1}^{J_{b,k}} w_{bk}[j]\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_{bk}[j],\boldsymbol{\Sigma}_{bk}[j]) \tag{6-58}$$

式中：$J_{\gamma k}$ 和 J_{bk} 分别为目标新生和目标衍生密度的高斯分量总数，由于高斯混合形式是一个强度函数而不是一个概率分布，所以其权值之和表示目标数目。

由此可得预测 PHD 函数为

$$D_{k|k-1}(\boldsymbol{x})=D_{s,k|k-1}(\boldsymbol{x})+D_{b,k|k-1}(\boldsymbol{x})+\gamma_k(\boldsymbol{x}) \tag{6-59}$$

式中：$D_{s,k|k-1}(\boldsymbol{x})$ 和 $D_{b,k|k-1}(\boldsymbol{x})$ 分别表示存在目标和衍生目标的 PHD。

$$\begin{aligned} D_{s,k|k-1}(\boldsymbol{x})&=P_{s,k}\sum_{i=1}^{J_{k-1}} w_{k-1}[i]\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_{s,k|k-1}[i],\boldsymbol{\Sigma}_{s,k|k-1}[i]) \\ &=\sum_{i=1}^{J_{k-1}} w_{s,k|k-1}[i]\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_{s,k|k-1}[i],\boldsymbol{\Sigma}_{s,k|k-1}[i]) \end{aligned} \tag{6-60}$$

$$\begin{aligned} D_{b,k|k-1}(\boldsymbol{x})&=\sum_{j=1}^{J_{b,k}}\sum_{i=1}^{J_{k-1}} w_{k-1}[i]w_{b,k}[j]\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_{k|k-1}[i,j],\boldsymbol{\Sigma}_{k|k-1}[i,j]) \\ &=\sum_{r=1}^{J_{b,k|k-1}} w_{k|k-1}[r]\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_{k|k-1}[r],\boldsymbol{\Sigma}_{k|k-1}[r]) \end{aligned} \tag{6-61}$$

对于式（6-54）和式（6-55）所示的系统方程，式（6-60）和式（6-61）中的预测状态及其协方差矩阵的具体形式取决于特定的求解算法。若采用基于局部线性化的 EKF 算法（采用其他算法如 CDKF、UKF 或 QKF 等算法流程与此类似），则

$$\begin{aligned} &\boldsymbol{\mu}_{s,k|k-1}[i]=f(\boldsymbol{\mu}_{k-1}[i]) \\ &w_{s,k|k-1}[i]=P_{s,k}w_{k-1}[i] \\ &\boldsymbol{\Sigma}_{s,k|k-1}[i]=\boldsymbol{F}_{k-1}[i]\boldsymbol{\Sigma}_{k-1}[i]\boldsymbol{F}_{k-1}^{\mathrm{T}}[i]+\boldsymbol{Q}_k \\ &\boldsymbol{\mu}_{k|k-1}[i,j]=\boldsymbol{\mu}_{k-1}[i]+\boldsymbol{\mu}_{b,k}[j] \\ &\boldsymbol{\Sigma}_{k|k-1}[i,j]=\boldsymbol{\Sigma}_{k-1}[i]+\boldsymbol{\Sigma}_{b,k}[j] \\ &\boldsymbol{F}_{k-1}[i]=\partial f/\partial \boldsymbol{x}\big|_{x=\boldsymbol{\mu}_{k-1}[i]} \end{aligned}$$

于是，预测 PHD 可进一步写成

$$D_{k|k-1}(\boldsymbol{x})=\sum_{i=1}^{J_{k|k-1}} w_{k|k-1}[i]\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_{k|k-1}[i],\boldsymbol{\Sigma}_{k|k-1}[i])$$

式中

$$J_{k|k-1}=J_{b,k|k-1}+J_{\gamma,k}+J_{k-1}$$

在更新阶段，其 PHD 可记为

$$D_k(\boldsymbol{x})=(1-P_D)D_{k|k-1}(\boldsymbol{x})+\sum_{\boldsymbol{Z}\in\boldsymbol{Z}_k} D_{L,k}(\boldsymbol{Z}|\boldsymbol{x})$$

$$D_{L,k}(\boldsymbol{Z}|\boldsymbol{x})=\sum_{j=1}^{J_{k|k-1}} w_k[j]\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_k[j],\boldsymbol{\Sigma}_k[j])$$

式中

$$w_k[j]=\frac{P_D w_{k|k-1}[j]g(z|\boldsymbol{\mu}_{k|k-1}[j])}{\lambda_c c_k(z)+\sum_{l=1}^{J_{k|k-1}} w_{k|k-1}[l]g(z|\boldsymbol{\mu}_{k|k-1}[l])}$$

$$g(z|\boldsymbol{\mu}_{k|k-1}[j])=\mathcal{N}(z;h(\boldsymbol{\mu}_{k|k-1}[j]),\boldsymbol{S}_k[j])$$

$$\boldsymbol{\mu}_k[j]=\boldsymbol{\mu}_{k|k-1}[j]+\boldsymbol{K}_k[j][z-h(\boldsymbol{\mu}_{k|k-1}[j])]$$

$$\boldsymbol{\Sigma}_k[j]=(\boldsymbol{I}_{n_x}-\boldsymbol{K}_k[j]\boldsymbol{H}_k[j])\boldsymbol{\Sigma}_{k|k-1}[j]$$

$$\boldsymbol{K}_k[j]=\boldsymbol{\Sigma}_{k|k-1}[j]\left(\boldsymbol{H}_k[j]\right)^{\mathrm{T}}\left(\boldsymbol{S}_k[j]\right)^{-1}$$

$$\boldsymbol{S}_k[j]=\boldsymbol{R}_k+\boldsymbol{H}_k[j]\boldsymbol{\Sigma}_{k|k-1}[j]\left(\boldsymbol{H}_k[j]\right)^{\mathrm{T}}$$

$$\boldsymbol{H}_k[j]=\partial h/\partial \boldsymbol{x}\big|_{\boldsymbol{x}=\boldsymbol{\mu}_{k|k-1}[j]}$$

由此可知，经 PHD 更新后，k 时刻总的高斯分量个数为 $J_k=(1+|\boldsymbol{Z}_k|)J_{k|k-1}$（$|\boldsymbol{Z}_k|$ 表示集合 $\boldsymbol{Z}_k$ 的势，即元素的个数）。这样，后验概率密度的高斯项随时间变化将会无限制地增加，为了解决这一问题，需要采取前文所提的剪枝、合并等措施使其控制在一定的数目范围之内。

经过剪枝、合并后，将高斯分量的权重进行累加得到目标数目的估计 $\hat{N}_k$，由此可通过提取权值最大的前 $\hat{N}_k$ 个高斯分量的均值作为目标状态的估计；也可通过将所有高斯分量的均值进行聚类的方法（如 k-mean 聚类等）得到 $\hat{N}_k$ 个聚类中心，从而完成目标状态的提取。

下面仍将以二维平面内的跟踪为例，验证 GM-PHD 算法在多目标条件下的跟踪能力。目标跟踪的系统模型如式（5-16）和式（5-19）所示，假设传感器位于坐标原点，采样周期 T=1s，检测概率 P_D=0.99，二维监视区域为 $[200\ 2000]\mathrm{m}\times[0\ 2]\pi$，距离和方位观测精度分别为 σ_r=2m，σ_φ=0.2°。初始时刻共有两个目标，其真实状态为 $\boldsymbol{x}_0^{(1)}=[250\ 2\ 250\ -10]^{\mathrm{T}}$，$\boldsymbol{x}_0^{(2)}=[-250\ 10\ -250\ 0]^{\mathrm{T}}$，过程噪声强度为 q=0.2m/s^2。在 k=21 时刻，由目标 1 所处的位置衍生一新目标，其初始速度为 $(-10,2)$m/s，3 个目标运动持续到 k=100 时刻，整个过程中无新生目标。目标航迹如图 6.6 所示。

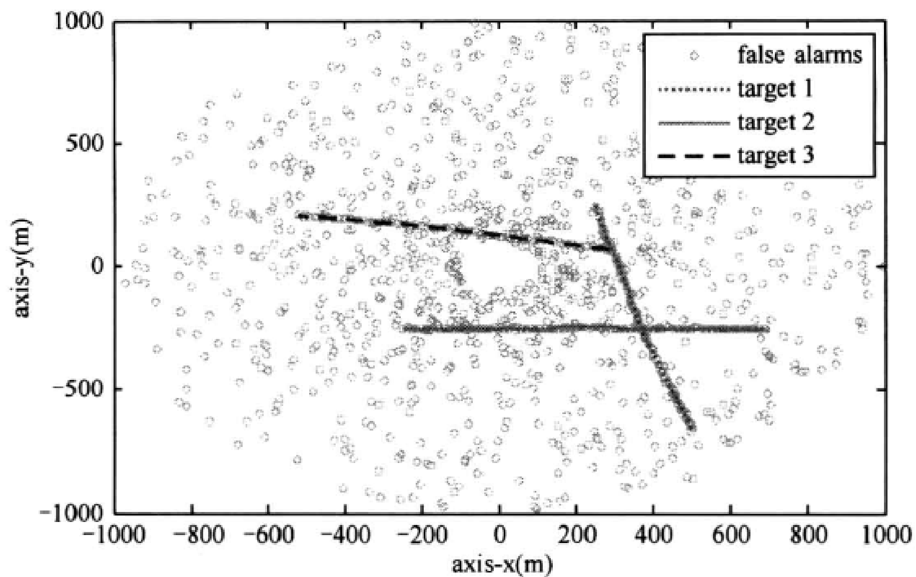


图 6.6　多目标航迹及虚警

假设目标的存活概率 P_s=0.99，每次扫描的平均虚警个数 λ_c=10，新生目标的强度为

$$\gamma(\boldsymbol{x})=0.1\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}[1],\boldsymbol{\Sigma})+0.1\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}[2],\boldsymbol{\Sigma})$$

式中

$$\boldsymbol{\mu}[1]=[250\ 0\ 250\ 0]^{\mathrm{T}}；\ \boldsymbol{\mu}[2]=[-250\ 0-250\ 0]^{\mathrm{T}}；\ \boldsymbol{\Sigma}=\mathrm{diag}([100\ 25\ 100\ 25])$$

衍生目标的强度为

$$b(\boldsymbol{x})=0.05\mathcal{N}(\boldsymbol{x};\xi,\boldsymbol{Q}_b)$$

式中：$\boldsymbol{Q}_b=\mathrm{diag}([100\ 400\ 100\ 400])$；$\xi$ 为目标衍生时的状态。

仿真中，假定 $\xi=[0\ 0\ 0\ 0]^{\mathrm{T}}$ ，每一时刻最大高斯分量个数为 $J_{\max}=20$ 。采用 GM-EKF 算法对目标进行跟踪，高斯分量的剪枝规则同式（6-13），其中门限 $d_{ij}=4$ ，所得结果如图 6.7~图 6.8 所示。

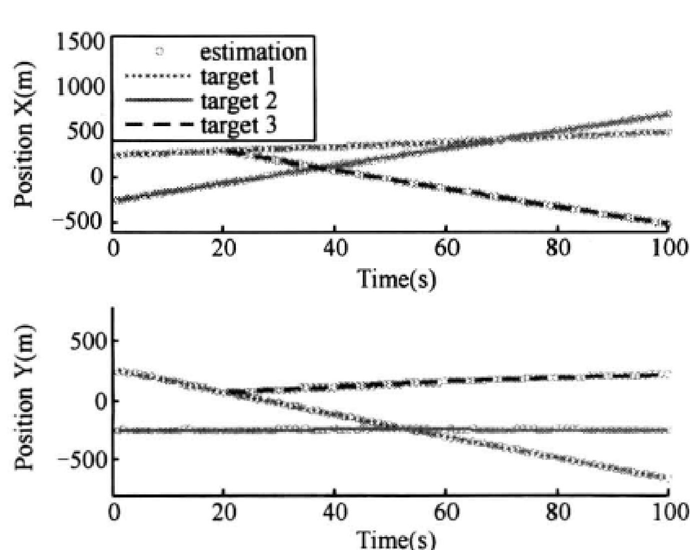


图 6.7　目标位置估计结果

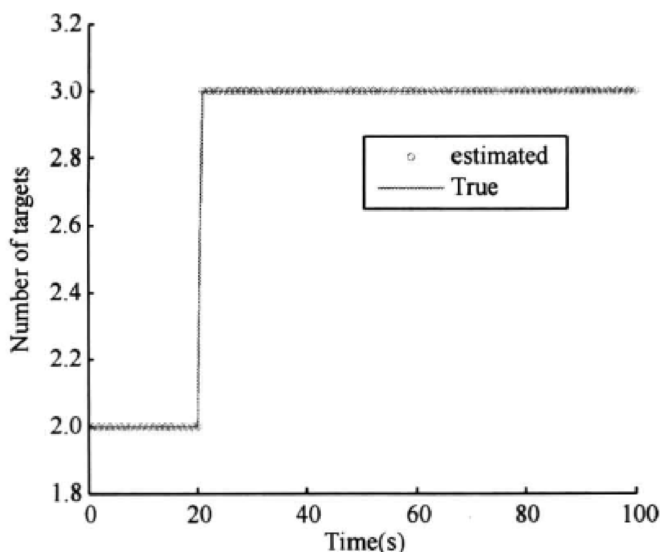


图 6.8　目标数目估计结果

由图 6.7 和图 6.8 中的结果可以看出，在多目标且目标个数可变的情况下，采用 GM-PHD 算法可以得到目标个数的正确估计，并实现对目标状态的准确提取。同时，从整个算法的处理流程可以看出，在处理过程中并没有用到传统多目标跟踪中的数据关联算法。这就说明，GM-PHD 算法在处理多目标跟踪问题时不仅是有效的，而且具有免数据关联的优势。

6.4　本 章 小 结

作为单高斯分量近似的扩展，混合高斯近似方法为非线性、非高斯条件下的滤波问题提供了一条有效的途径。在现实世界中，系统的非高斯性可能体现在过程噪声、观测噪声或系统状态等单一因素或几种因素的综合，在这种情况下，单高斯分量已不能对其进行有效近似，这一点在闪烁噪声条件下的目标跟踪结果中已得到充分体现。与此不同的是，混合高斯分布具有以任意精度逼近非高斯分布的能力，由此得到的混合高斯滤波算法（如 GM-EKF、GM-UKF 等）自然能够适应非高斯条件下的滤波问题。在多目标条件下，目标的存在、消失、衍生/新生等变化过程同样可建模为高斯混合分布，以此为基础的高斯混合形式成为概率假设密度（PHD）滤波的一条重要手段，在时变多目标跟踪应用中发挥着重要作用。

参 考 文 献

[1] Bar-shalom Y，X R Li，Kirubarajan T. Estimation with Application to Tracking and Navigation：Theory，Algorithm，and Software[M]. New York：John Wiley & Sons，2001.

[2] Bell B M，Cathey F W. The iterated Kalman filter update as a Gauss-Newton method[J]. IEEE Transactions on Automatic Control，1993，38（2）：294–297.

[3] Galkowski P G，Islam M A. An alternative derivation of the modified gain function of Song and Speyer[J]. IEEE Transaction on Automatic Control，1991，36（11）：1323–1326.

[4] Guerci J R，Goetz R，Dimodica J. A method for improving extended kalman filter performance for angle-only passive ranging[J]. IEEE Transactions on Aerospace and Electronic Systems，1994，30（4）：1090–1093.

[5] Fucheng G，Zhongkang S，Kan H. A modified covariance extended Kalman filtering algorithm in passive location[C]. Proceedings of the IEEE International Conference on Robotics，Intelligent Systems and Signal Processing，2003：307–311.

[6] Dahlquist G，Bjöck A. Numerical Methods[M]. New Jersey：Prentice-Hall，1974.

[7] NØgaard M，Poulsen N，Ravn O. Advances in derivative-free state estimation for nonlinear systems[R]. Technical Report，Technical University of Denmark，2000.

[8] van der Merwe R，Wan E. Efficient derivative-free Kalman filters for online learning[C]. Proceedings of the 9th European Symposium on Artificial Neural Networks，2001：205–210.

[9] Daowang F，Teng L，Tao H Z. Square-root second-order extended Kalman filter and its application in target motion analysis[J]. IET Radar，Sonar and Navigation，2010，4（3）：329–335.

[10] Levent O，Aliev F A. Coment on adaptive fading kalman filter with an application[J]. Automatica，1998，34（12）：1663–1664.

[11] D H Zhou，Frank P M. Strong tracking filtering of nonlinear time-varying stochastic systems with coloured noise：application to parameter estimation and empirical robustness analysis[J]. International Journal Control，1996，65（2）：295–307.

[12] 刘福声，罗鹏飞. 统计信号处理[M]. 长沙：国防科技大学出版社，1999.

[13] Julier S J，Uhlmann J K，Durrant-whyte H F. A new approach for filtering nonlinear systems[C]. Proceedings of the American Control Conference，1995：1628–1632.

[14] Julier S J，Uhlmann J K. A new method for the nonlinear transformation of means and covariances in filters and estimators[J]. IEEE Transactions on Automatic Control，2000，45（3）：477–482.

[15] Julier S J，Uhlmann J K. Unscented filtering and nonlinear estimation[J]. Proceedings of the IEEE，2004，92（3）：401–422.

[16] Lefebvre T，Bruyninclcx H，De Schutter J. Comment on "A new method for the nonlinear transformation of means and covariances in filters and estimators"[J]. IEEE Transactions on Automatic Control，2002，47（8）：1406–1408.

[17] Julier S J. The scaled unscented transformation[C]. Proceedings of American Control Conference，2002：4555–4559.

[18] Tenne D，Singh T. The higher order unscented filter[C]. Proceedings of American Control Conference，2003：2441–2446.

[19] Van De Merwe R，Wan E A. The square-root unscented Kalman filter for state and parameter estimation[C]. Proceedings of IEEE International Conference on Acoustics，Speech，and Signal Processing，2001：3461–3464.

[20] Bell B M，Cathey F W. The iterated Kalman filter update as a Gauss-Newton method[J]. IEEE Transactions on Automatic Control，1993，38（2）：294–297.

[21] 龚享铱. 利用频率变化率和波达角变化率单站无源定位与跟踪的关键技术研究[D]. 长沙：国防科学技术大学博士学位论文，2004.

[22] Johnston L A，Krishnamurthy V. Derivation of a sawtooth iterated extended Kalman smoother via the AECM algorithm[J]. IEEE Transactions on Signal Processing，2001，49（9）：1899–1909.

[23] Ronghui Z，Jianwei W. Iterated unscented Kalman filter for passive target tracking[J]. IEEE Transactions on Aerospace and

Electronic Systems，2007，43（3）：1155–1162.

[24] 占荣辉. 基于空频域信息的单站被动目标跟踪算法研究[D]. 长沙：国防科学技术大学博士学位论文，2007.

[25] 占荣辉，郁春来，万建伟. 一种简化的 UKF 算法及其在单站无源目标跟踪中的应用[J]. 现代雷达，2007，29（3）：42–46.

[26] 占荣辉，郁春来，万建伟. 基于角度和频率信息的机动目标无源自适应跟踪算法[J]. 信号处理，2007，23（6）：853–856.

[27] Arasaratnam I，Haykin S，Elliott R J. Discrete-time nonlinear filtering algorithms using Gauss–Hermite quadrature[J]. Proceedings of the IEEE，2007，95（5）：953–976.

[28] Arasaratnam I，Haykin S. Square-root quadrature Kalman filtering[J]. IEEE Transactions on Signal Processing，2008，56（6）：2589–2593.

[29] Arasaratnam I，Haykin S. Cubature Kalman filters[J]. IEEE Transactions on Automatic Control，2009，54（6）：1254–1269.

[30] Anderson B D O，Moore J B. Optimal Filtering[M]. New Jersey：Prentice-Hall，1979.

[31] Christopher M B. Pattern Recognition and Machine Learning[M]. New York：Springer，2006.

[32] Mahler R P S. Statistical Multisource-Multitarget Information Fusion[M]. London：Arteth House，2006.

[33] Mahler R. Multi-target Bayes filtering via first-order multi-target moments[J]. IEEE Transactions on Aerospace and Electronic Systems，2003，39（4）：1152–1178.

[34] B-N Vo，W-K Ma. The Gaussian mixture probability hypothesis density filter[J]. IEEE Transactions on Signal Processing，2006，54（11）：4091–4104.

[35] 张俊根，姬红兵. 高斯混合粒子 PHD 滤波被动测角多目标跟踪[J]. 控制与决策，2011，26（3）：413–417.

[36] 郝燕玲，孟凡彬，王素鑫，等. 一种新的多机动目标跟踪的 GMPHD 滤波算法[J]. 上海交通大学学报，2011，44（7）：873–877.

第三部分　随机采样近似滤波与应用

与第二部分阐述的基于解析高斯近似的方法不同，另一类递推型非线性滤波算法通过追踪系统状态的条件概率分布来实现状态的估计，并通过贝叶斯定理实现条件概率的转移[1]。这类算法的近似对象不是状态函数或观测函数，而是状态的条件概率密度，其目的是近似真实的状态分布而不是非线性函数本身，且用于近似的采样样本是随机的而不是确定性的。由于这类算法建立在蒙特卡罗仿真技术基础之上，通常称为序贯蒙特卡罗（Sequential Monte Carlo，SMC）法[2-4]或“粒子”滤波法（Particle Filtering，PF）[5-8]。在这类方法中，通过蒙特卡罗仿真手段产生大量的粒子，并由其散布情况来逼近状态的概率分布。

由于粒子滤波通过非参数化的蒙特卡罗模拟方法来实现递推贝叶斯估计，因此适用于任何能用状态空间模型表示的非线性系统，以及传统解析高斯滤波近似算法难以表示的非高斯系统，且精度可以逼进最优估计[9]。从这一点意义来说，解析高斯近似方法能适应的问题，粒子滤波的方法自然也能适应。此外，传统的解析近似高斯滤波方法（如EKF）对初值选择较为敏感，若初值选择不当可能会降低收敛速度或者导致滤波器发散。相比之下，粒子滤波器却有其独特的优势，由于粒子的散布性，很容易在一定的误差范围内快速捕获到真实的状态，从而提高跟踪系统的稳定性和收敛速度。

但是，一般情况下，用蒙特卡罗仿真方法实现递推滤波其运算量要远大于解析高斯近似方法，这在某种程度上影响了其实时应用。不过，粒子滤波实现方便且适合并行处理，随着数字处理技术的快速发展该方法具有广阔的应用前景。

本部分主要阐述粒子滤波方法及其在目标跟踪中的应用问题。第 7 章首先从粒子滤波的基本理论入手，阐述标准粒子滤波算法的实现方法和存在的主要问题。第 8 章在此基础上研究了基于重要采样函数的粒子滤波算法。针对非线性系统中存在线性子系统这一类重要混合系统，出于降低算法复杂度和提高滤波精度等目的，第 9 章专门介绍状态分布的“边缘化”思想及由此得到的边缘化粒子滤波算法。第 10 章主要从提高粒子产生质量的角度出发，总结和分析了几种基于智能优化思想的粒子采样原理和粒子滤波实现方法。

第 7 章　先验分布采样滤波

在非线性条件下，贝叶斯滤波面临一个重要问题是状态分布的表达和积分式的求解，由前面章节中的分析可知，对于一般的非线性/非高斯系统，解析求解的途径是行不通的。在数值近似方法中，蒙特卡罗仿真是一种最为通用、有效的手段，粒子滤波就是建立在蒙特卡罗仿真基础之上的，它通过利用一组带权值的系统状态采样来近似状态的统计分布。由于蒙特卡罗仿真方法具有广泛的适用性，由此得到的粒子滤波算法也能适用于一般的非线性/非高斯系统。但是，这种滤波方法也面临几个重要问题，如有效采样（粒子）如何产生、粒子如何传递以及系统状态的序贯估计如何得到等。

本章首先从贝叶斯滤波的随机采样近似一般原理出发，介绍了重要采样技术及其序贯处理方法；在此基础上针对一种应用最广、处理最方便的特殊最优采样函数，得到了标准粒子滤波（SPF）算法的处理流程，给出了应用实例。

7.1　贝叶斯滤波的随机采样近似

假定存在如下非线性系统模型

$$\boldsymbol{x}_{k+1}=f(\boldsymbol{x}_k,\boldsymbol{w}_k) \tag{7-1}$$

$$\boldsymbol{y}_k=h(\boldsymbol{x}_k,\boldsymbol{v}_k) \tag{7-2}$$

式中：$\boldsymbol{x}_k$ 为系统状态变量；$\boldsymbol{y}_k$ 为观测；$\boldsymbol{w}_k$ 和 $\boldsymbol{v}_k$ 分别表示过程噪声和观测噪声。

贝叶斯方法求解状态估计问题是通过用已获得的观测量来建立当前状态 $\boldsymbol{x}_k$ 的后验 PDF 来实现的，一旦获得 $p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k})$，也就得到了关于状态 $\boldsymbol{x}_k$ 某一映射函数 $g(\cdot)$ 的估计：

$$I(g(\boldsymbol{x}_k))=E_{p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})}[g(\boldsymbol{x}_k)]=\int g(\boldsymbol{x}_k)p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k})\mathrm{d}\boldsymbol{x}_k \tag{7-3}$$

一般情况下，对上式进行解析求解是比较困难的，只能得到某种近似解。根据蒙特卡罗仿真原理，该积分算式可近似为

$$\hat{E}_{p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})}[g(\boldsymbol{x}_k)]\approx\frac{1}{N}\sum_{i=1}^{N}g(\boldsymbol{x}_k^{(i)}) \tag{7-4}$$

式中：$\{\boldsymbol{x}_k^{(i)},i=1,2,\cdots,N\}$ 表示从后验分布函数 $p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k})$ 采样得到的 N 个独立同分布样本。当 N 充分大时，根据大数定理，其期望的估计“几乎总是”（almost surely）收敛于真实的期望，即

$$\hat{E}_{p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})}[g(\boldsymbol{x}_k)]\xrightarrow[a.s.]{N\to\infty}E_{p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})}[g(\boldsymbol{x}_k)] \tag{7-5}$$

同时，若 $g(\boldsymbol{x}_k)$ 的后验方差有界，即

$$\mathrm{var}[g(\boldsymbol{x}_k)]=E[g(\boldsymbol{x}_k)\cdot g(\boldsymbol{x}_k)^{\mathrm{T}}]<\infty \tag{7-6}$$

则由中心极限定理[4]可得

$$\sqrt{N}(\hat{E}[g(\boldsymbol{x}_k)]-E[g(\boldsymbol{x}_k)])\overset{N\to\infty}{\Rightarrow}\mathcal{N}(\boldsymbol{0},\mathrm{var}[g(\boldsymbol{x}_k)]) \tag{7-7}$$

式中：$\overset{N\to\infty}{\Rightarrow}$ 表示“依分布收敛于”。

需要指出的是，从后验滤波分布 $p(\boldsymbol{x}_k\mid\boldsymbol{y}_{1:k})$ 进行采样近似只是蒙特卡罗仿真技术的一种特殊形式，实际上更一般的情况是从完整的后验分布 $p(\boldsymbol{x}_{0:k}\mid\boldsymbol{y}_{1:k})$ 中进行采样。

7.2 重要采样技术

如前所述，对于一般的非线性系统，很难求得后验分布的解析解，也就无法直接从 $p(\boldsymbol{x}_{0:k}\mid\boldsymbol{y}_{0:k})$ 进行采样；一种可替代的方案是通过重要采样（Importance Sampling，IS）技术从某一已知的、易于采样的函数进行间接采样，这种函数通常称为重要密度函数（Importance Density Function，IDF），其分布称为建议分布（Proposal distribution）。暂不考虑 IDF 的具体形式，假设满足要求的函数为 $\pi(\boldsymbol{x}_{0:k}\mid\boldsymbol{y}_{1:k})$，则由重要采样原理[10-11]，有

$$\begin{aligned}I[f(\boldsymbol{x}_{0:k})]&=\int f(\boldsymbol{x}_{0:k})\frac{p(\boldsymbol{x}_{0:k}\mid\boldsymbol{y}_{1:k})}{\pi(\boldsymbol{x}_{0:k}\mid\boldsymbol{y}_{1:k})}\pi(\boldsymbol{x}_{0:k}\mid\boldsymbol{y}_{1:k})\mathrm{d}\boldsymbol{x}_{0:k}\\&=\int f(\boldsymbol{x}_{0:k})\frac{p(\boldsymbol{y}_{1:k}\mid\boldsymbol{x}_{0:k})p(\boldsymbol{x}_{0:k})}{\pi(\boldsymbol{x}_{0:k}\mid\boldsymbol{y}_{1:k})p(\boldsymbol{y}_{1:k})}\pi(\boldsymbol{x}_{0:k}\mid\boldsymbol{y}_{1:k})\mathrm{d}\boldsymbol{x}_{0:k}\\&=\int f(\boldsymbol{x}_{0:k})\frac{w_k(\boldsymbol{x}_{0:k})}{p(\boldsymbol{y}_{1:k})}\pi(\boldsymbol{x}_{0:k}\mid\boldsymbol{y}_{1:k})\mathrm{d}\boldsymbol{x}_{0:k}\end{aligned} \tag{7-8}$$

式中：$w_k(\boldsymbol{x}_{0:k})$ 称为重要权，其值为

$$w_k(\boldsymbol{x}_{0:k})=\frac{p(\boldsymbol{y}_{1:k}\mid\boldsymbol{x}_{0:k})p(\boldsymbol{x}_{0:k})}{\pi(\boldsymbol{x}_{0:k}\mid\boldsymbol{y}_{1:k})} \tag{7-9}$$

为方便起见，通常将 $w_k(\boldsymbol{x}_{0:k})$ 简记为 w_k，式（7-8）进一步可化为

$$\begin{aligned}I[f(\boldsymbol{x}_{0:k})]&=\frac{1}{p(\boldsymbol{y}_{1:k})}\int f(\boldsymbol{x}_{0:k})w_k(\boldsymbol{x}_{0:k})\pi(\boldsymbol{x}_{0:k}\mid\boldsymbol{y}_{1:k})\mathrm{d}\boldsymbol{x}_{0:k}\\&=\frac{\int f(\boldsymbol{x}_{0:k})w_k(\boldsymbol{x}_{0:k})\pi(\boldsymbol{x}_{0:k}\mid\boldsymbol{y}_{1:k})\mathrm{d}\boldsymbol{x}_{0:k}}{\int p(\boldsymbol{y}_{1:k}\mid\boldsymbol{x}_{0:k})p(\boldsymbol{x}_{0:k})\dfrac{\pi(\boldsymbol{x}_{0:k}\mid\boldsymbol{y}_{1:k})}{\pi(\boldsymbol{x}_{0:k}\mid\boldsymbol{y}_{1:k})}\mathrm{d}\boldsymbol{x}_{0:k}}\\&=\frac{\int f(\boldsymbol{x}_{0:k})w_k(\boldsymbol{x}_{0:k})\pi(\boldsymbol{x}_{0:k}\mid\boldsymbol{y}_{1:k})\mathrm{d}\boldsymbol{x}_{0:k}}{\int w_k(\boldsymbol{x}_{0:k})\pi(\boldsymbol{x}_{0:k}\mid\boldsymbol{y}_{1:k})\mathrm{d}\boldsymbol{x}_{0:k}}=\frac{I_\pi[f(\boldsymbol{x}_{0:k})w_k(\boldsymbol{x}_{0:k})]}{I_\pi[w_k(\boldsymbol{x}_{0:k})]}\end{aligned} \tag{7-10}$$

即假定从建议分布 $\pi(\boldsymbol{x}_{0:k}\mid\boldsymbol{y}_{0:k})$ 进行采样得到一组粒子 $\{\boldsymbol{x}_{0:k}^{(i)},i=1,2,\cdots,N\}$，则后验均值的估计

$$\hat{I}[f(\boldsymbol{x}_{0:k})]=\frac{\frac{1}{N}\sum_{i=1}^{N}f(\boldsymbol{x}_{0:k}^{(i)})w_k(\boldsymbol{x}_{0:k}^{(i)})}{\frac{1}{N}\sum_{i=1}^{N}w_k(\boldsymbol{x}_{0:k}^{(i)})}=\sum_{i=1}^{N}\tilde{w}_k^{(i)}f(\boldsymbol{x}_{0:k}^{(i)}) \tag{7-11}$$

其中，归一化权值 $\tilde{w}_k^{(i)}$ 可表示为

$$\tilde{w}_k^{(i)} = \frac{w_k(\boldsymbol{x}_{0:k}^{(i)})}{\sum_{i=1}^{N} w_k(\boldsymbol{x}_{0:k}^{(i)})} \tag{7-12}$$

7.3 序贯重要采样

若选择的 IDF 可分解为

$$\pi(\boldsymbol{x}_{0:k} \mid \boldsymbol{y}_{1:k}) = \pi(\boldsymbol{x}_k \mid \boldsymbol{x}_{0:k-1}, \boldsymbol{y}_{1:k})\pi(\boldsymbol{x}_{0:k-1} \mid \boldsymbol{y}_{1:k-1}) \tag{7-13}$$

且系统模型满足以下三点假设：

（1）系统状态服从一阶马尔可夫过程。

（2）不同时刻的系统观测相互独立。

（3）初始的先验密度为 $p(\boldsymbol{x}_0)$。

即

$$p(\boldsymbol{x}_{0:k}) = p(\boldsymbol{x}_0)\prod_{j=1}^{k} p(\boldsymbol{x}_j \mid \boldsymbol{x}_{j-1}) \tag{7-14}$$

$$p(\boldsymbol{y}_{1:k} \mid \boldsymbol{x}_{0:k}) = \prod_{j=1}^{k} p(\boldsymbol{y}_j \mid \boldsymbol{x}_j) \tag{7-15}$$

则可得出重要权的递推计算式为

$$\begin{aligned} w_k &= \frac{p(\boldsymbol{y}_{1:k} \mid \boldsymbol{x}_{0:k})p(\boldsymbol{x}_{0:k})}{\pi(\boldsymbol{x}_k \mid \boldsymbol{x}_{0:k-1}, \boldsymbol{y}_{1:k})\pi(\boldsymbol{x}_{0:k-1} \mid \boldsymbol{y}_{1:k-1})} \\ &= \frac{p(\boldsymbol{y}_{1:k} \mid \boldsymbol{x}_{0:k})p(\boldsymbol{x}_{0:k})p(\boldsymbol{y}_{1:k-1} \mid \boldsymbol{x}_{0:k-1})p(\boldsymbol{x}_{0:k-1})}{\pi(\boldsymbol{x}_k \mid \boldsymbol{x}_{0:k-1}, \boldsymbol{y}_{1:k})\pi(\boldsymbol{x}_{0:k-1} \mid \boldsymbol{y}_{1:k-1})p(\boldsymbol{y}_{1:k-1} \mid \boldsymbol{x}_{0:k-1})p(\boldsymbol{x}_{0:k-1})} \\ &= w_{k-1}\frac{p(\boldsymbol{y}_{1:k} \mid \boldsymbol{x}_{0:k})p(\boldsymbol{x}_{0:k})}{\pi(\boldsymbol{x}_k \mid \boldsymbol{x}_{0:k-1}, \boldsymbol{y}_{1:k})p(\boldsymbol{y}_{1:k-1} \mid \boldsymbol{x}_{0:k-1})p(\boldsymbol{x}_{0:k-1})} \\ &= w_{k-1}\frac{p(\boldsymbol{y}_k \mid \boldsymbol{x}_k)p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1})}{\pi(\boldsymbol{x}_k \mid \boldsymbol{x}_{0:k-1}, \boldsymbol{y}_{1:k})} \end{aligned} \tag{7-16}$$

特别地，如果 IDF 满足

$$\pi(\boldsymbol{x}_k \mid \boldsymbol{x}_{0:k-1}, \boldsymbol{y}_{1:k}) = \pi(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}, \boldsymbol{y}_k) \tag{7-17}$$

则由该函数采样得到的粒子 $\boldsymbol{x}_k^{(i)} \sim \pi(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}, \boldsymbol{y}_k)$ 建立的后验滤波概率密度可近似为

$$\hat{p}(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k}) = \sum_{i=1}^{N} \tilde{w}_k^{(i)}\delta(\boldsymbol{x}_k - \boldsymbol{x}_k^{(i)}) \tag{7-18}$$

$$\hat{E}[f(\boldsymbol{x}_k)] = \sum_{i=1}^{N} \tilde{w}_k^{(i)} f(\boldsymbol{x}_k)\delta(\boldsymbol{x}_k - \boldsymbol{x}_k^{(i)}) = \sum_{i=1}^{N} \tilde{w}_k^{(i)} f(\boldsymbol{x}_k^{(i)}) \tag{7-19}$$

图 7.1 给出了用粒子及相应权 $\left\{\boldsymbol{x}_k^{(i)}, w_k^{(i)}\right\}_{i=1}^{N}$ 表示后验分布 $\hat{p}(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k})$ 的示例，图中圆圈的大小代表粒子的权值，权值越大表示该粒子在 PDF 的比重越大（即对整个 PDF 的贡献也大）；反之亦然。

文献[12]和文献[13]已经证明，权值的方差会随时间递推而增加，因此，序贯重要采样（SIS）粒子滤波算法有一个无法避免的粒子退化（Degeneracy）问题[14]。主要表现为经若干次递推以后，除了少量几个粒子以外，大部分粒子的权值小至可以忽略。粒子退

化程度可用有效样本个数 N_{eff} （Effective Sample Size）来度量，其定义为

$$N_{\text{eff}} = \frac{N}{1+\text{var}\{\tilde{w}_k^{(i)}\}} \tag{7-20}$$

N_{eff} 越小，说明退化现象越严重。

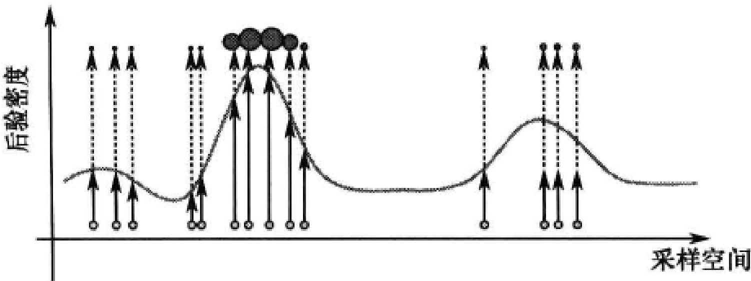


图 7.1　用粒子表示的后验分布

通常，有两种方法可以减轻粒子退化问题：一是增加重采样环节；二是选择合适的 IDF 进行更有效的采样。下面分别加以介绍。

7.3.1　粒子重采样

减轻粒子退化的方法之一是加入重采样环节。重采样的基本思想是消除权值较小的粒子而对权值显著的粒子进行多份复制，其方法是对后验密度的离散近似

$$\hat{p}(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k}) = \sum\nolimits_{i=1}^{N} \tilde{w}_k^{(i)} \delta(\boldsymbol{x}_k - \boldsymbol{x}_k^{(i)}) \tag{7-21}$$

再进行一次采样，重新生成一组新的粒子 $\{\boldsymbol{x}_k^{(j)}, j=1,2,\cdots,N\}$ 并使其满足

$$\Pr\{\boldsymbol{x}_k^{(j)} = \boldsymbol{x}_k^{(i)}\} = \tilde{w}_k^{(i)} \tag{7-22}$$

这一过程可以通过图 7.2 加以解释，同样图中圆圈的大小代表所占的权重。

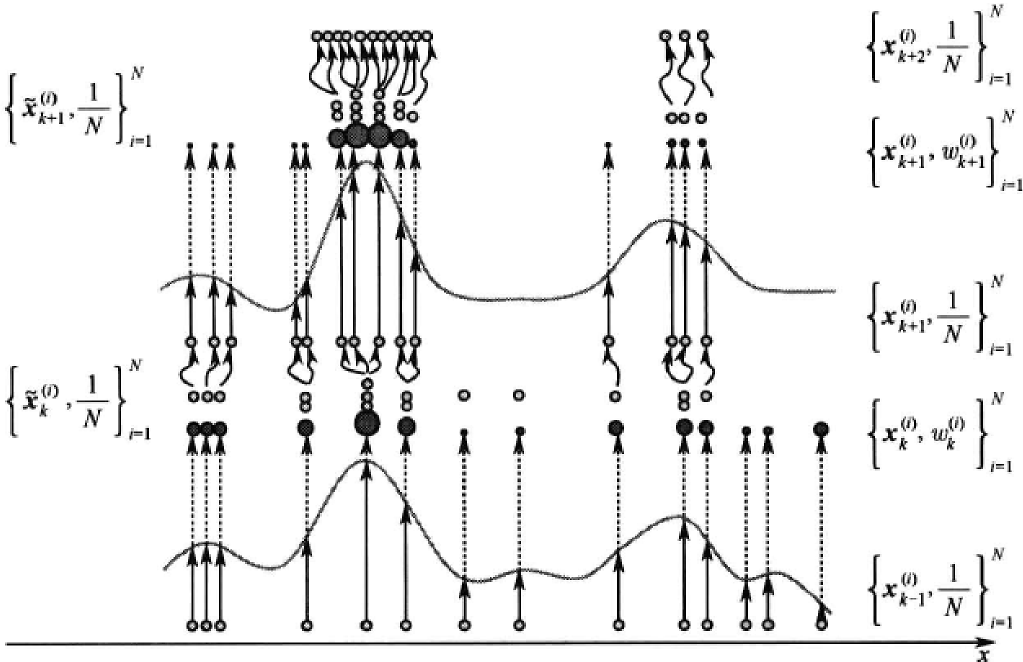


图 7.2　重采样过程示意

在 $k-1$ 时刻 N 个粒子具有相等的权值，通过序贯采样计算，k 时刻粒子的权值发生了变化。通过重采样环节，权值大的粒子得到多次复制（选择），权值小的粒子被忽略（淘

汰）。由于新的粒子独立同分布，重采样后粒子的权值均为 $w_k^{(i)}=1/N$ 。重采样过程避免了将大量的计算耗费在那些只占有微小量权的粒子上（对 PDF 的贡献几乎可以忽略），降低了权值的方差。常见的重采样方法有层次采样（Stratified Sampling）[15]、残差采样（Residual Sampling）[16]及系统采样（Systemic Sampling）[17]等。

设离散随机变量 X 的分布为概率累积形式

$$F(x)=P(X\leqslant x)=\sum_{x^i<x}p(x_i)$$

将 [0,1] 区间分成 N 个子区间 $\left(0,p(x^1)\right]$，$\left(p(x^1),p(x^1)+p(x^2)\right)$，…，$\left(\sum_{m=1}^{N-1}p(x^m),\sum_{m=1}^{N}p(x^m)\right]$。设 u 是[0,1]区间上的均匀分布随机变量，根据 u 的值落在哪个区间，相应区间对应的随机变量即为所需输出量。假设落在第 j 个区间，则输出为 x^j，粒子的子代数 n^i 表示 u 的值落在该区间的次数。如图 7.3 所示给出了重采样的实现过程。

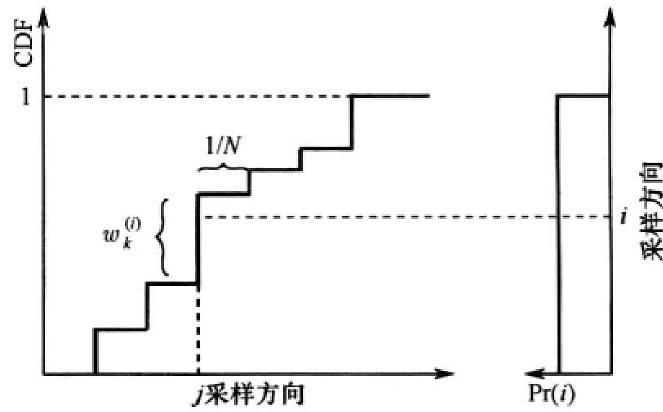


图 7.3　重采样过程图解

作为例子，下面将给出系统重采样的流程。

$$[\{x_k^{(j)},w_k^{(j)},i^{(j)}\}_{j=1}^N]=\text{SystematicResample}[\{x_k^{(i)},w_k^{(i)}\}_{i=1}^N]$$

步骤 1：初始化累积分布函数 CDF：$c_0=0$。

步骤 2：构造 CDF：$c_i(i=1,2,\cdots,N)$。

$$c_i=c_{i-1}+w_k^{(i)}$$

步骤 3：由 CDF 的底部开始启动 $i=1$，并抽取起始点 $u_1\sim\mathcal{U}[0,1/N]$。

步骤 4：沿 CDF 移动：$u_j=u_1+(j-1)/N\ (j=1,2,\cdots,N)$。

当 $u_j>c_i$ 时，$i=i+1$。

$$x_k^{(j*)}=x_k^{(i)}$$

$$w_k^{(j)}=1/N$$

$$i^{(j)}=i$$

必须指出的是，虽然重采样在一定程度上可以减轻粒子退化现象，但却因此带来了采样贫化（Sample Impoverishment）问题。所谓采样贫化，就是经重采样以后，原本的粒子集合中有很多粒子（由于权值太小）没有“子代”；而少数几个粒子则有很多相同的“子代”，丧失了粒子的多样性。为此，人们提出了许多不同的方法来解决这个问题，如采用

重采样粒子移动算法（Resample-Move Algorithm）[18]、增加马尔可夫链蒙特卡罗（Markov Chain Monte Carlo）[19-21]移动步骤、对粒子进行正则（Regularization）重采样等[4]。

7.3.2 重要密度函数

减轻粒子退化的另一种策略是选择合适的 IDF。文献[14]中指出，当选择的 IDF $\pi(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}, \boldsymbol{y}_k) = p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}, \boldsymbol{y}_k)$ 时，可使权值的方差最小化。这是因为，在此条件下，有

$$\operatorname{var}_\pi\{w_k^{(i)}\} = (w_{k-1}^{(i)})^2 \left[\int \frac{(p(\boldsymbol{y}_k \mid \boldsymbol{x}_k) p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}^{(i)}))^2}{\pi(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{y}_{1:k})} \mathrm{d}\boldsymbol{x}_k - p^2(\boldsymbol{y}_k \mid \boldsymbol{x}_{k-1}^{(i)}) \right] = 0 \tag{7-23}$$

显然，当 $\pi(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}, \boldsymbol{y}_k) = p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}, \boldsymbol{y}_k)$ 时，对于任意给定的粒子 $\boldsymbol{x}_{k-1}^{(i)}$，无论怎样从 $\pi(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}, \boldsymbol{y}_k)$ 进行采样，$w_k^{(i)}$ 都有同样的取值。即

$$\begin{aligned} w_k^{(i)} &= w_{k-1}^{(i)} \frac{p(\boldsymbol{y}_k \mid \boldsymbol{x}_k) p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}^{(i)})}{p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{y}_k)} = w_{k-1}^{(i)} \frac{p(\boldsymbol{y}_k \mid \boldsymbol{x}_k) p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}^{(i)}) p(\boldsymbol{y}_k \mid \boldsymbol{x}_{k-1}^{(i)})}{p(\boldsymbol{y}_k \mid \boldsymbol{x}_k) p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}^{(i)})} \\ &= w_{k-1}^{(i)} p(\boldsymbol{y}_k \mid \boldsymbol{x}_{k-1}^{(i)}) = w_{k-1}^{(i)} \int p(\boldsymbol{y}_k \mid \boldsymbol{x}_k) p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}^{(i)}) \mathrm{d}\boldsymbol{x}_k \end{aligned} \tag{7-24}$$

的计算不依赖于 $\boldsymbol{x}_k^{(i)}$。需要说明的是，这里的方差指的是由不同的采样 $\boldsymbol{x}_k^{(i)}$ 所导致的 $w_k^{(i)}$ 的方差。

最优（权值方差最小意义下）IDF 要求从 $p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{y}_k)$ 进行采样，同时需要计算积分项 $\int p(\boldsymbol{y}_k \mid \boldsymbol{x}_k) p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}^{(i)}) \mathrm{d}\boldsymbol{x}_k$。在通常情况下，很难求得该积分项的解析表达式，因此实际应用中往往采用便于实现的近似最优函数来代替。

7.4 标准粒子滤波算法

前面提到，当采用最优采样函数时，算法实现比较困难。一种最简单、常用的替代方案是选择先验 $p(\boldsymbol{x}_{k+1} \mid \boldsymbol{x}_k)$ 作为 IDF，此时

$$\pi(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{y}_k) = p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}^{(i)}) \tag{7-25}$$

通过对该函数进行采样得到的粒子 $\boldsymbol{x}_k^{(i)} \sim p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}^{(i)})$ 其权值为

$$w_k^{(i)} = w_{k-1}^{(i)} p(\boldsymbol{y}_k \mid \boldsymbol{x}_k^{(i)}) \tag{7-26}$$

由此便得到如下的滤波算法及图 7.4 所示的实现结构图。

Alg（7.1）：Standard Particle Filtering（SPF）

（1）由 $p(\boldsymbol{x}_0)$ 得到 N 个采样 $\{\boldsymbol{x}_0^{(i)}\}_{i=1}^N$。

（2）计算权值 $w_k^{(i)} = p(\boldsymbol{y}_k \mid \boldsymbol{x}_k^{(i)})$ 并对其归一化，即 $\tilde{w}_k^{(i)} = w_k^{(i)} / \sum_{i=1}^N w_k^{(i)}$。

（3）对粒子进行重采样，使其满足 $\Pr(\boldsymbol{x}_k^{(j)} = \boldsymbol{x}_k^{(i)}) = \tilde{w}_k^{(i)}$。

（4）时间更新：根据状态转移函数产生新的粒子 $\boldsymbol{x}_{k+1}^{(i)} \sim p(\boldsymbol{x}_{k+1} \mid \boldsymbol{x}_k^{(i)})$。

（5）重复步骤（2）～步骤（4）。

最后得到 k 时刻状态量的估计为

$$\hat{\boldsymbol{x}}_k = \sum_{i=1}^N \tilde{w}_k^{(i)} \boldsymbol{x}_k^{(i)} \approx E_{p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k})}[\boldsymbol{x}_k] \tag{7-27}$$

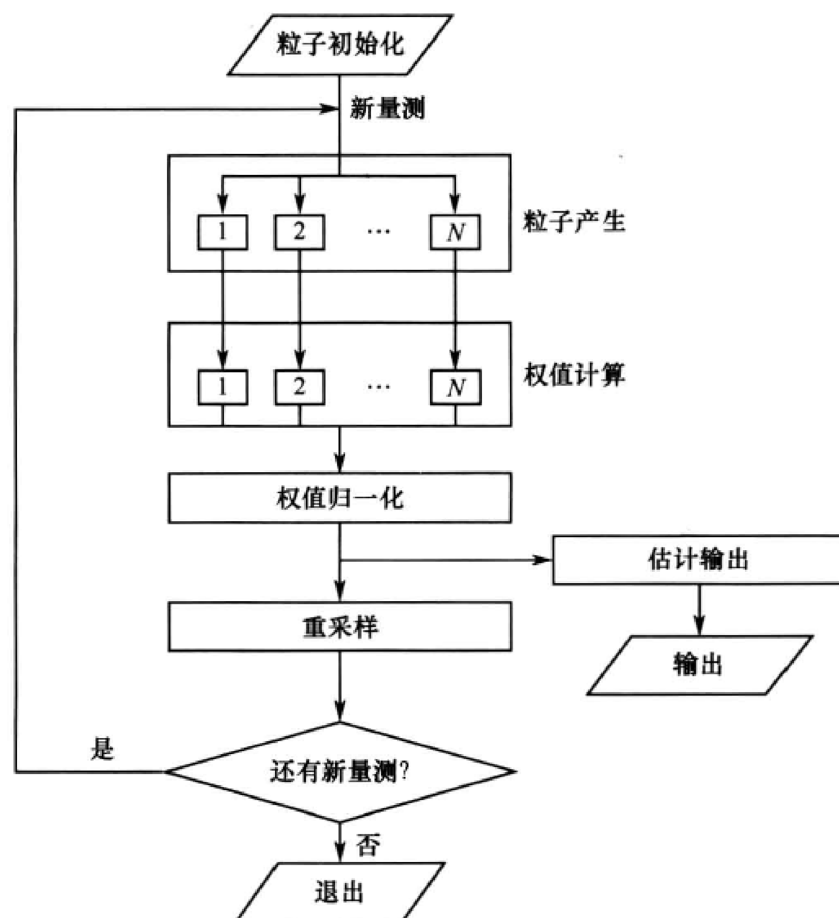


图 7.4 粒子滤波算法实现结构

对应于 Alg（7.1）中的处理流程，图 7.4 给出了直观的算法实现结构。为便于表述，这里将通过先验分布（Prior）产生粒子得到的滤波算法称为标准粒子滤波（SPF）算法。

下面通过一典型例子来说明粒子滤波算法的工作过程。假定某非线性函数的状态方程为

$$x_k=0.5x_{k-1}+\frac{25x_{k-1}}{1+x_{k-1}^2}+8\cos[1.2(k-1)]+w_k \tag{7-28}$$

观测模型为

$$y_k=x_k^2/20+v_k \tag{7-29}$$

式中：w_k 和 v_k 均为零均值高斯噪声且 $w_k \sim \mathcal{N}(0,Q)$，$v_k \sim \mathcal{N}(0,R)$。该系统具有明显的高度非线性，且似然函数呈双峰特征，这些特点使得传统滤波方法在估计系统状态时显得非常困难。

假设系统初始状态 $x_0=0$，$Q=2$，$R=1$，根据上述系统方程产生 40 个时刻状态数据和观测数据，如图 7.5 和图 7.6 所示。初始时刻的粒子通过高斯分布函数产生，即 $x_0^{(i)} \sim \mathcal{N}(0,2)$，$(i=1,2,\cdots,N)$。采用 N=2000 个粒子进行估计，所得结果如图 7.7 和图 7.8 所示。

在算法实现过程中，假定有效粒子数 $N_{eff}<0.5N$ 时进行重采样处理。由此可见，采用 SMC 方法并结合重采样技术可以得到构建出系统的后验状态分布，从而得到系统状态量的有效估计。从算法的实现上看，整个过程中采用的是系统状态 PDF 近似的方法，不涉及对非线性系统函数本身的近似。这就说明，尽管本实例中采用的是高斯系统噪声，但粒子滤波方法适应用一般的非线性/非高斯条件——这正是粒子滤波算法的特色所在。

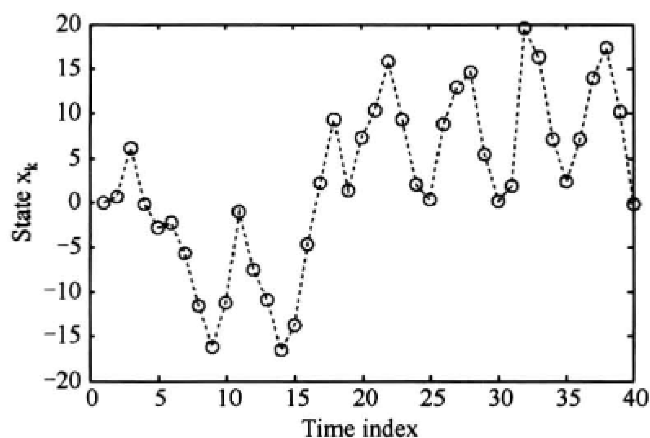


图 7.5 系统状态序列

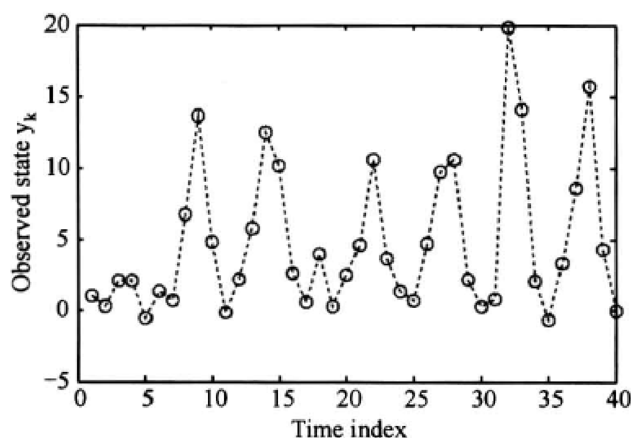


图 7.6 系统观测序列

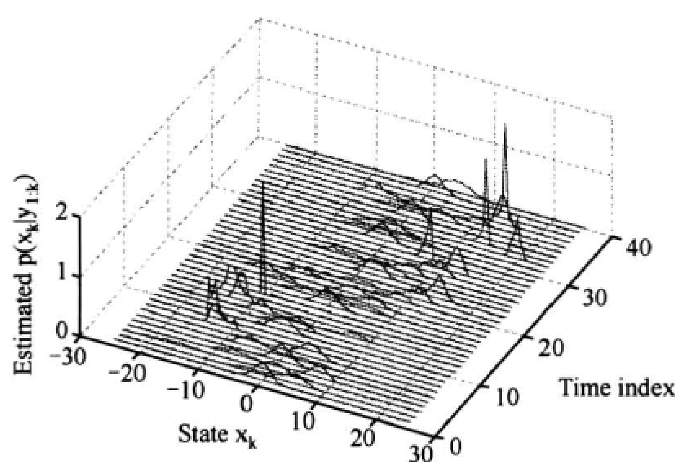


图 7.7 后验分布估计

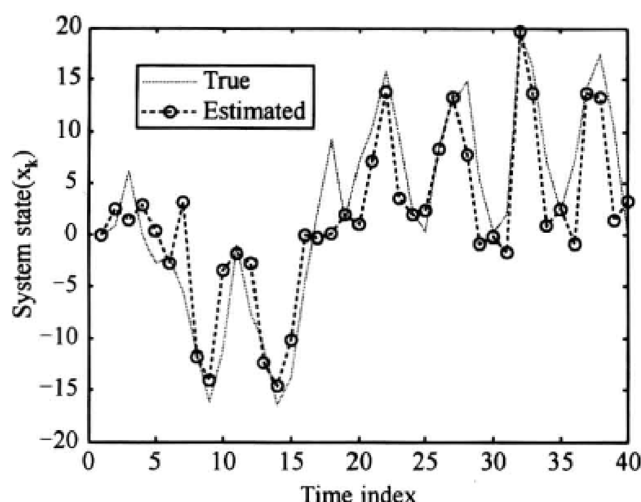
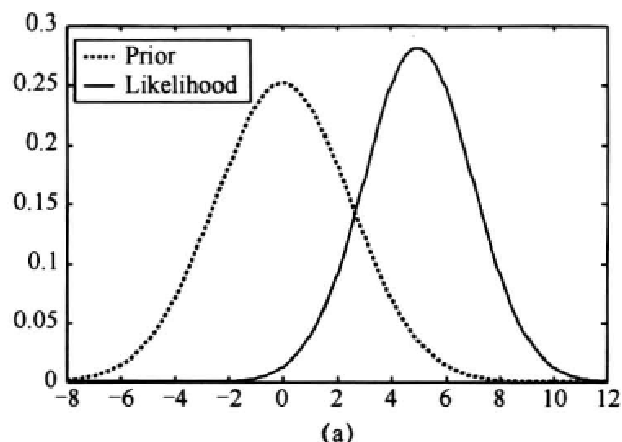


图 7.8 系统状态估计

SPF 最大的优点在于算法简单且易于实现，其缺点是没有融入新的观测值。此时，从 IDF 中所取的样本与由真实的后验分布产生的样本存在较大的偏差，当似然函数位于先验的尾部或者观测模型具有很高的精度时（图 7.9），这种偏差尤为明显[22]。这样，从 IDF 得到的大部分粒子其权值都可以忽略（即对 PDF 的贡献几乎为零），大量的计算将浪费于此，效率极其低下。



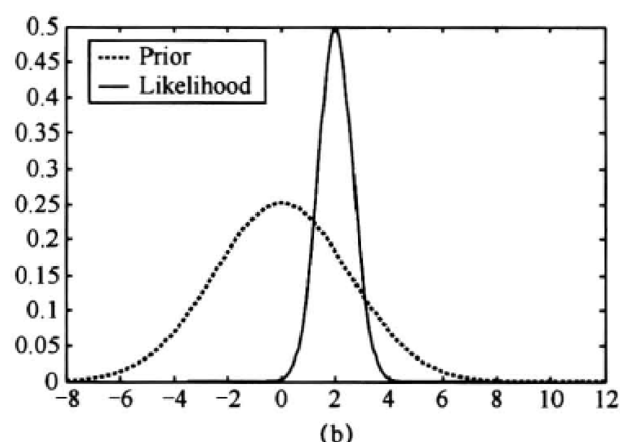


图 7.9 似然函数与先验位置关系

为此，人们提出各种改进措施，如采用正则采样方法、辅助粒子滤波采样方法[23]、EKF 采样方法和 UKF 采样方法[24]等，详见第 8 章。

7.5 应用实例

7.5.1 雷达制导跟踪

以 x–y 二维平面内的制导跟踪应用为例，假定目标状态为 $\boldsymbol{x}_k^t=[x^t\ \dot{x}^t\ y^t\ \dot{y}^t]^{\mathrm{T}}(k)$，目标的运动方程满足

$$\boldsymbol{x}_k^t=\boldsymbol{F}\boldsymbol{x}_{k-1}^t+\boldsymbol{w}_k^t \tag{7-30}$$

式中

$$\boldsymbol{F}=\begin{bmatrix}\boldsymbol{F}_1 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{F}_1\end{bmatrix},\quad \boldsymbol{F}_1=\begin{bmatrix}1 & T\\ 0 & 1\end{bmatrix} \tag{7-31}$$

T 为采样间隔；$\boldsymbol{w}_k^t$ 为目标运动的过程噪声，这里假定服从零均值高斯分布，其协方差矩阵为

$$\boldsymbol{Q}_k=q^2\cdot\begin{bmatrix}\boldsymbol{Q}_1 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{Q}_1\end{bmatrix},\quad \boldsymbol{Q}_1=\begin{bmatrix}T^3/3 & T^2/2\\ T^2/2 & T\end{bmatrix} \tag{7-32}$$

导弹的运动方程为

$$\boldsymbol{x}_k^m=\boldsymbol{F}\boldsymbol{x}_{k-1}^m+\boldsymbol{G}\boldsymbol{u}_{k-1}+\boldsymbol{w}_k^m$$

式中

$$\boldsymbol{x}_k^m=[x^m\ \dot{x}^m\ y^m\ \dot{y}^m]^{\mathrm{T}}(k)$$

$$\boldsymbol{G}_1=\begin{bmatrix}-T^2/2\\ -T\end{bmatrix},\quad \boldsymbol{G}=\begin{bmatrix}\boldsymbol{G}_1 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{G}_1\end{bmatrix},\quad \boldsymbol{u}=\begin{bmatrix}a_x\\ a_y\end{bmatrix}$$

$$a_x=[c_1\ c_2]\begin{bmatrix}x_k\\ \dot{x}_k\end{bmatrix},\quad a_y=[c_1\ c_2]\begin{bmatrix}y_k\\ \dot{y}_k\end{bmatrix}$$

$\boldsymbol{x}_k=\boldsymbol{x}_k^t-\boldsymbol{x}_k^m$ 为相对状态；$c_1=N_{\mathrm{g}}/t_{\mathrm{go}}^2$，$c_2=N_{\mathrm{g}}/t_{\mathrm{go}}$，$N_{\mathrm{g}}$ 为导引系数，t_{go} 为剩余时间；$\boldsymbol{w}_k^m$ 为导弹运动的过程噪声，同样可建模为服从零均值高斯分布。

传感器位于弹载平台观测坐标系原点，其观测量包括目标的距离和方位角，观测方程可写为

$$\boldsymbol{y}_k=h(\boldsymbol{x}_k)+\boldsymbol{v}_k=\begin{bmatrix}\sqrt{x_k^2+y_k^2}\\ \arctan(y_k/x_k)\end{bmatrix}+\begin{bmatrix}v_{r,k}\\ v_{\varphi,k}\end{bmatrix} \tag{7-33}$$

式中：距离和方位观测噪声 $v_{r,k}$、$v_{\varphi,k}$ 为非高斯闪烁噪声。

本实例中，假设闪烁噪声为高斯分布与拉普拉斯分布的混合形式，即

$$\begin{aligned}p(v_k)&=(1-\varepsilon)p_1(v_k)+\varepsilon p_2(v_k)\\ &=(1-\varepsilon)\mathcal{N}(v_k;\mu_1,\sigma_1^2)+\varepsilon\mathcal{L}(v_k;\mu_2,\sigma_2^2)\end{aligned} \tag{7-34}$$

式中：$\mathcal{L}(v_k;\mu_2,\sigma_2^2)=\dfrac{1}{\sqrt{2}\sigma_2}\exp\left(-\sqrt{2}\left|v_k-\mu_2\right|/\sigma_2\right)$ 为拉普拉斯函数；ε 为闪烁概率。

设 $\mu_1=\mu_2=0$，$\sigma_1=1$，$\sigma_2=5$，$\varepsilon=0.2$，得到的闪烁噪声序列如图 7.10 所示，图中拉普拉斯噪声采用如下方法得到，即

$$v_L=\sigma_L \frac{v_1 v_2+v_3 v_4}{\sqrt{2}} \tag{7-35}$$

式中：v_1、v_2、v_3和v_4为相互独立的标准正态分布随机数；σ_L为拉普拉斯噪声标准差。

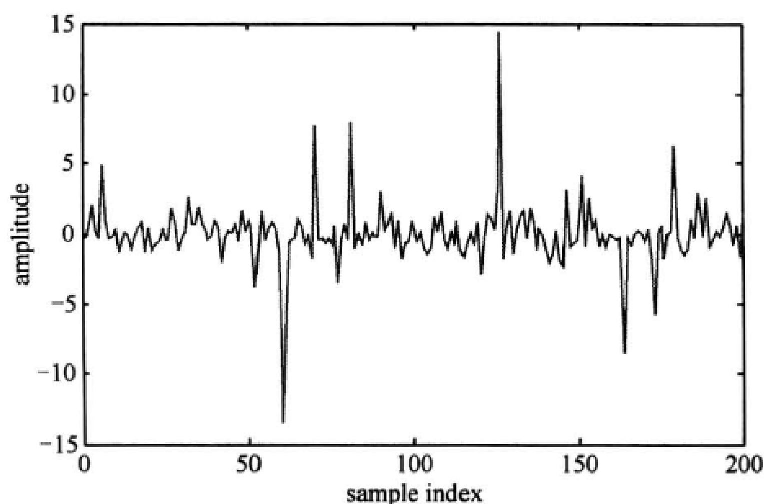


图 7.10　闪烁噪声序列

仿真中假定目标初始位置和速度分别为(20,1.5)km和(−100,50)m/s，过程噪声强度为q^t=2m/s^2；导弹初始位置为(40,15)km，初始速度为(−1000,−150)m/s，为方便起见，不考虑过程噪声的影响。进一步假定弹目总的交会时间为T_f=61s，时间采样间隔T=0.25s，导引系数N_g=3，$t_{k,\text{go}}=T_\text{f}-k\cdot T(k=1,2,\cdots)$，图 7.11 给出了前 60s 弹目接近的过程。

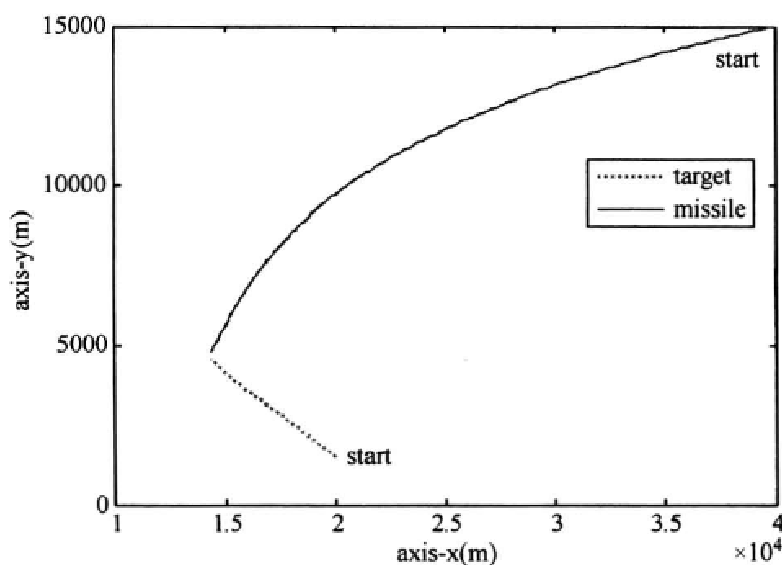


图 7.11　弹目交会场景

距离和方位观测噪声模型满足式（7-34），式中σ_{r_1}=20m，σ_{φ_1}=0.2°，$\sigma_{r_2}=5\sigma_{r_1}$，$\sigma_{\varphi_1}=5\sigma_{\varphi_2}$，$\varepsilon$=0.25。采用标准粒子滤波（PF）算法和 EKF 算法分别对目标进行跟踪，初始误差协方差矩阵$\boldsymbol{P}_0$=diag([50^2 10^2 50^2 10^2])，粒子数M=2000。实验次数为 200，所得结果如图 7.12～图 7.16 所示。

图 7.12 给出了单次跟踪的示例（局部放大结果），图 7.13～图 7.16 为两种算法的位置均方根误差和速度均方根误差比较。由图中的结果可以看出，在高斯+拉普拉斯混合噪声条件下，PF 得到的结果要好于 EKF，这是因为 PF 采用的是概率密度近似方法，可以

用粒子表征非高斯分布噪声；而 EKF 采用的是矩（后验均值和方差）近似法，实例中采用的协方差矩阵为 $\boldsymbol{R}=(1-\varepsilon)\boldsymbol{R}_1+\varepsilon\boldsymbol{R}_2$（$\boldsymbol{R}_1$、$\boldsymbol{R}_2$ 分别为高斯和拉普拉斯观测噪声协方差），无法充分描述真实噪声的统计特性，因而存在较大的估计误差。另外，从图中的结果可以发现，较之 EKF，PF 算法在初始跟踪建立后收敛速度更快。这是因为 PF 算法在初始化时采用了大量的粒子，其中有部分粒子散布在目标的真实状态附近，这些粒子将占有较大的权重，从而快速引导其他粒子向目标的真实状态附近聚集，使跟踪误差迅速降低。

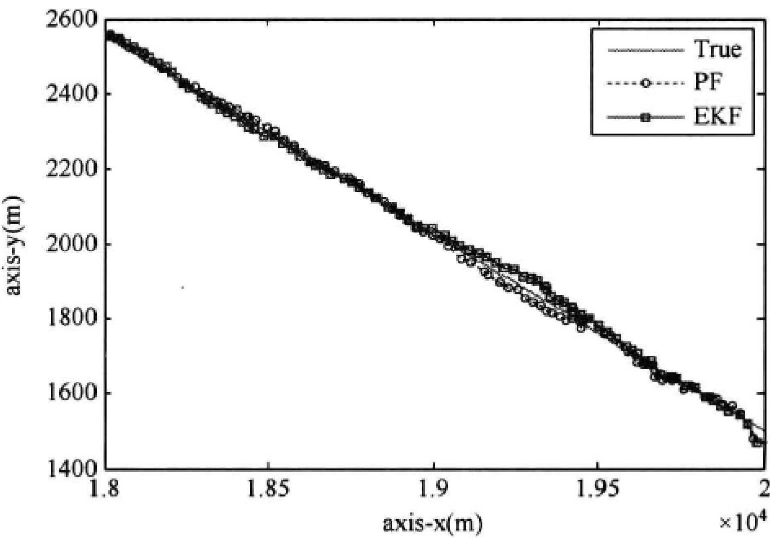


图 7.12　单次跟踪示例

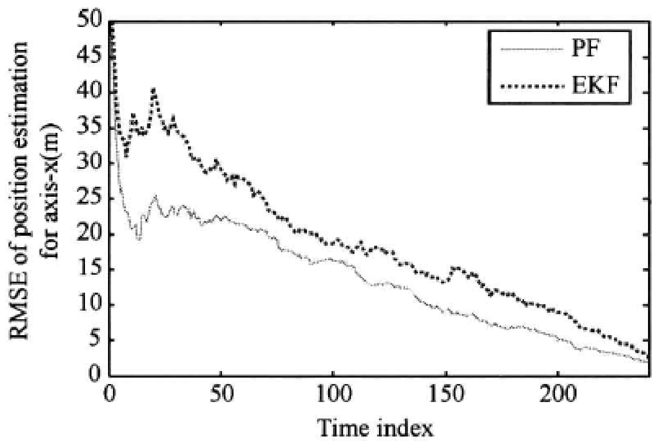


图 7.13　x 轴位置估计均方根误差

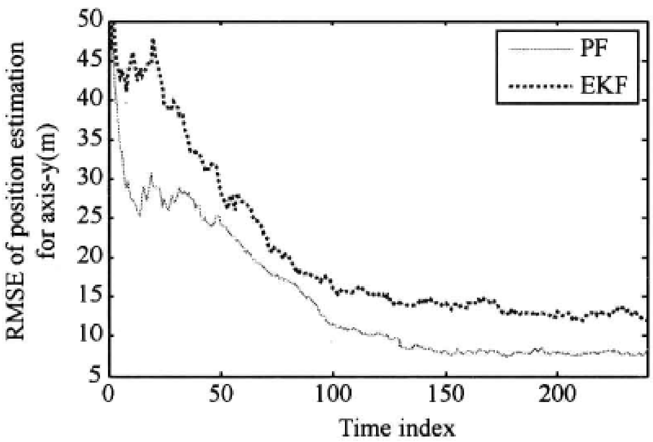


图 7.14　y 轴位置估计均方根误差

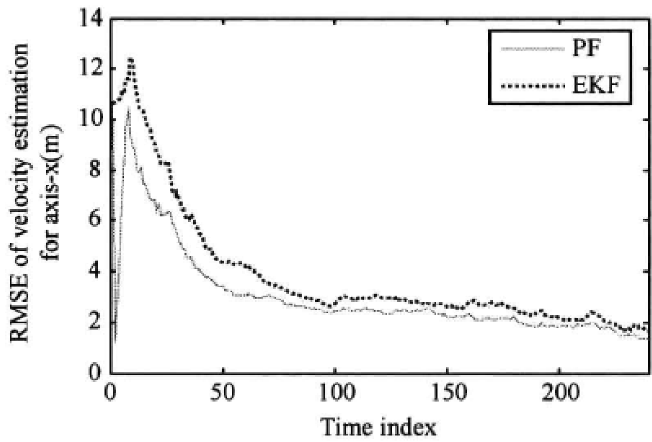


图 7.15　x 轴速度估计均方根误差

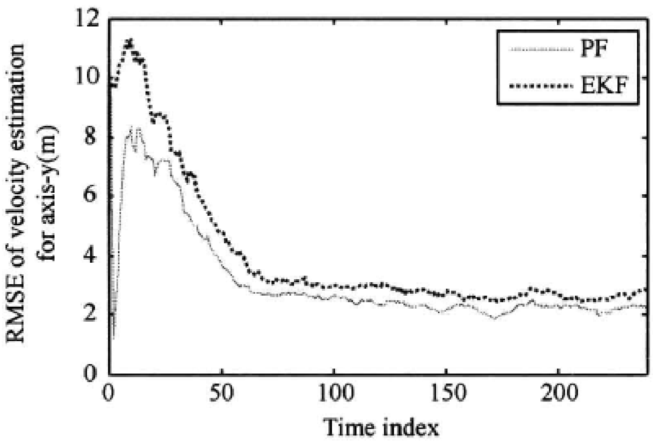


图 7.16　y 轴速度估计均方根误差

两种算法的时间复杂度见表 7.1，由此可以看出 PF 的算法复杂度要比 EKF 大得多，这主要是因为粒子个数较多，在实现上没有做到完全并行；同时，算法中的重采样环节也会导致时间消耗增大。

表 7.1　两种算法运行时间比较

执行时间 \ 算法	EKF	PF
Mean/（s/run）	0.0105	0.3595
STD/（s/run）	2.73e-4	0.0304

为了进一步分析粒子数目对跟踪性能的影响，图 7.17 给出了 M=1000、2000、4000、8000 条件下、第 20 个采样时刻（5s）以后的平均均方根误差（ARMSE）。由图中的结果可以看出，随着粒子数目的增加，估计性能也相应有了改善；不过当粒子数目大到一定程度以后（如 M>4000），增加粒子数对提高估计效果的作用不再明显，但算法的复杂度还是不断增大的，如图 7.18 所示。因此，实际应用时，应综合考虑性能和复杂度等因素，选择适中的粒子规模。

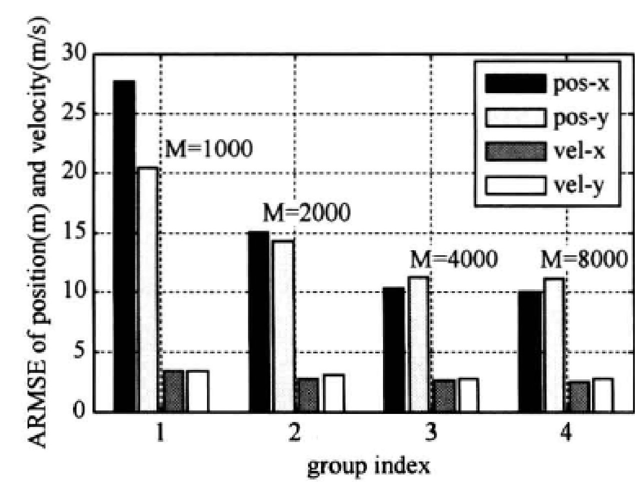


图 7.17　不同粒子数目条件下的跟踪性能

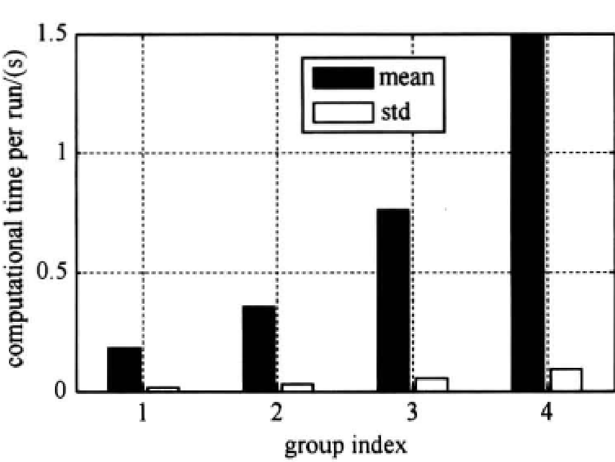


图 7.18　不同粒子数目条件下的时间消耗

7.5.2　SMC-PHD 多目标跟踪

在第 6 章曾提到，混合高斯近似形式为实现 PHD 滤波的一条重要途径，本例子中将介绍 PHD 滤波的另一种重要途径——基于 SMC 的实现方法及在多目标检测前跟踪的应用。

考虑多目标应用场景，假设 N_k 个目标的状态演化方程可表示为

$$\boldsymbol{x}_k^s = f_k(\boldsymbol{x}_{k-1}^s, \boldsymbol{w}_k), \quad s = 1,2,\cdots,N_k \tag{7-36}$$

式中：$\boldsymbol{x}_k^s = [x_k^s \ \dot{x}_k^s \ y_k^s \ \dot{y}_k^s \ I_k^s]^{\mathrm{T}}$ 为 k 时刻目标 s 的状态，包含目标位置、速度和信号强度等信息；f_k 表示一般的状态转移函数；$\boldsymbol{w}_k$ 为过程噪声。

传感器提供的二维监视区域（具有 $n_x \times n_y$ 个分辨单元）的量测可表示为

$$z_k^{(i,j)} = \begin{cases} \sum_{t=1}^{N_k} h_k^{(i,j)}(\boldsymbol{x}_k^s) + v_k^{(i,j)} & H_1 \\ v_k^{(i,j)} & H_0 \end{cases} \tag{7-37}$$

式中：$h_k^{(i,j)}(\boldsymbol{x}_k^s)$ 为观测方程，表示目标 s 的信号强度对网格单元 (i,j) 的贡献；$v(i,j)$ 为网格之间相互独立的观测噪声。

假定采用点扩展模型，目标强度可表示为

$$h_k^{(i,j)}(\boldsymbol{x}_k^s)=\frac{\Delta_x\Delta_y I_k^s}{2\pi\Sigma^2}\exp\left[-\frac{(i\Delta_x-x_k^s)^2+(j\Delta_y-y_k^s)^2}{2\Sigma^2}\right] \tag{7-38}$$

式中：Σ 为由传感器分辨率引起的模糊量。

k 时刻的量测集 z_k 及到 k 时刻为止的累积测量集 $z_{1:k}$ 可表示为

$$z_{1:k}=\{z_l:l=1,2,\cdots,k\} \tag{7-39}$$

式中：$\boldsymbol{z}_k=\{z_k^{(i,j)}:i=1,2,\cdots,n_x;j=1,2,\cdots,n_y\}$。

在独立噪声假设条件下，传感器的似然函数可表示为

$$p(\boldsymbol{z}_k\mid\boldsymbol{x}_k^1,\cdots,\boldsymbol{x}_k^{N_k})=\begin{cases}\prod\limits_{i=1}^{n_x}\prod\limits_{j=1}^{n_y}p_{S+N}(z_k^{(i,j)}\mid\boldsymbol{x}_k^1,\cdots,\boldsymbol{x}_k^{N_k}) & H_1\\ \prod\limits_{i=1}^{n_x}\prod\limits_{j=1}^{n_y}p_N(z_k^{(i,j)}) & H_0\end{cases} \tag{7-40}$$

式中：$p_N(z_k^{(i,j)})$ 表示网格单元 (i,j) 的背景噪声 PDF，$p_{S+N}(z_k^{(i,j)}\mid\boldsymbol{x}_k^1,\cdots,\boldsymbol{x}_k^{N_k})$ 表示目标+噪声的似然函数。

进一步假设背景噪声为高斯噪声，此时有

$$p_{S+N}(z_k^{(i,j)}\mid H_1)=\mathcal{N}(z_k^{(i,j)};\sum_{s=1}^{N_k}h_k^{(i,j)}(\boldsymbol{x}_k^s),\sigma^2) \tag{7-41}$$

$$p_N(z_k^{(i,j)}\mid H_0)=\mathcal{N}(z_k^{(i,j)};0,\sigma^2) \tag{7-42}$$

式中：σ^2 为噪声方差。

由于目标信号仅对部分网格单元产生影响，在这种条件下，似然函数 $p(\boldsymbol{z}_k\mid\boldsymbol{x}_k^t)$ 可近似为

$$p(\boldsymbol{z}_k\mid\boldsymbol{x}_k^s,H_1)\approx\prod_{i\in C_i(\boldsymbol{x}_k^s)}\prod_{j\in C_j(\boldsymbol{x}_k^s)}p_{S+N}(z_k^{(i,j)}\mid\boldsymbol{x}_k^s)\cdot\prod_{i\notin C_i(\boldsymbol{x}_k^s)}\prod_{j\notin C_j(\boldsymbol{x}_k^s)}p_N(z_k^{(i,j)}) \tag{7-43}$$

式中：$i_0=D_i(\boldsymbol{x}_k^s)$，$j_0=D_j(\boldsymbol{x}_k^s)$，$C_i(\boldsymbol{x}_k)=\{i_0-q,\cdots,i_0+q\}$，$C_j(\boldsymbol{x}_k)=\{j_0-q,\cdots,j_0+q\}$；$\left(D_i(\boldsymbol{x}_k^s),D_j(\boldsymbol{x}_k^s)\right)=\max\limits_{i,j}h_k^{(i,j)}(\boldsymbol{x}_k^s)$ 表示受目标强度影响最大的网格单元。

在 PHD 滤波框架下[25-27]，预测 PHD 方程为

$$D_{k|k-1}(\boldsymbol{x}_k\mid z_{1:k-1})=\int D_{k|k-1}(\boldsymbol{x}_k\mid\boldsymbol{x}_{k-1})D_{k-1}(\boldsymbol{x}_{k-1}\mid z_{1:k-1})\mathrm{d}\boldsymbol{x}_{k-1}+\gamma_k(\boldsymbol{x}_k) \tag{7-44}$$

$$D_{k|k-1}(\boldsymbol{x}_k\mid\boldsymbol{x}_{k-1})=b_{k|k-1}(\boldsymbol{x}_k\mid\boldsymbol{x}_{k-1})+e_{k|k-1}f_{k|k-1}(\boldsymbol{x}_k\mid\boldsymbol{x}_{k-1}) \tag{7-45}$$

式中：$\gamma_k(\boldsymbol{x}_k)$ 为新生目标；$e_{k|k-1}$ 为目标存活概率；$b_{k|k-1}(\boldsymbol{x}_k\mid\boldsymbol{x}_{k-1})$ 为从已存在目标衍生出的目标 PHD；$f_{k|k-1}(\boldsymbol{x}_k\mid\boldsymbol{x}_{k-1})$ 为转移概率密度函数。

在获得 k 时刻的量测 $\boldsymbol{z}_k$ 之后，更新方程可表示为

$$D_k(\boldsymbol{x}_k\mid z_{1:k})=\left[(1-P_d(\boldsymbol{x}_k))+\sum_{z\in\boldsymbol{z}_k}\frac{P_d(\boldsymbol{x}_k)g_k(z\mid\boldsymbol{x}_k)}{\lambda_k c_k(z)+D_k(z\mid z_{1:k-1})}\right]\cdot D_{k|k-1}(\boldsymbol{x}_k\mid z_{1:k-1}) \tag{7-46}$$

$$D_k(z \mid z_{1:k-1}) = \int P_d(\boldsymbol{x}_k) g_k(z \mid \boldsymbol{x}_k) D_{k|k-1}(\boldsymbol{x}_k \mid z_{1:k-1}) \mathrm{d}\boldsymbol{x}_k \tag{7-47}$$

式中：$P_d(\boldsymbol{x}_k)$表示检测概率；λ_k为每帧数据的虚警个数；$c_k(z)$为虚警的分布函数；$g_k(z \mid \boldsymbol{x}_k)$为目标的似然函数。

式（7-44）～式（7-47）构成了 PHD 滤波的递推过程，监视区域A中期望的目标数为

$$N_k = \int_A D_k(\boldsymbol{x}_k \mid z_{1:k}) \mathrm{d}\boldsymbol{x}_k$$

由系统方程式(7-36)和式(7-37)可知，式(7-46)中的似然函数可表示为$g_k(z \mid \boldsymbol{x}_k) = p_{S+N}(z \mid \boldsymbol{x}_k)$。若定义似然比为$l(z \mid \boldsymbol{x}_k) = p_{S+N}(z \mid \boldsymbol{x}_k) / p_N(z)$，易知$l(z \mid \boldsymbol{x}_k) \propto g_k(z \mid \boldsymbol{x}_k)$。在检测前跟踪（TbD）应用中，由于不设置门限，故有$P_d(\boldsymbol{x}_k) \equiv 1$。于是式（7-46）可等效为

$$D_k'(\boldsymbol{x}_k \mid z_{1:k}) = \left[\sum_{z \in z_k} \frac{l(z \mid \boldsymbol{x}_k)}{\lambda_k + D_k'(z \mid z_{1:k-1})} \right] D_{k|k-1}(\boldsymbol{x}_k \mid z_{1:k-1}) \tag{7-48}$$

$$D_k'(z \mid z_{1:k-1}) = \int l(z \mid \boldsymbol{x}_k) D_{k|k-1}(\boldsymbol{x}_k \mid z_{1:k-1}) \mathrm{d}\boldsymbol{x}_k \tag{7-49}$$

在非线性/非高斯条件下，上述表达式很难求得闭式解。作为一种数值近似手段，SMC 方法为解决这类问题提供了一条有效的手段。在基于 SMC-PHD 的 TbD 算法实现中，后验密度$D_{k-1}'(\boldsymbol{x}_{k-1} \mid z_{1:k-1})$可由一组带权值$\{w_{k-1}^{(p)}\}_{p=1}^{L_{k-1}}$的粒子集$\{\boldsymbol{x}_{k-1}^{(p)}\}_{p=1}^{L_{k-1}}$近似为

$$D_{k-1}'(\boldsymbol{x}_{k-1} \mid z_{1:k-1}) = \sum_{p=1}^{L_{k-1}} w_{k-1}^{(p)} \delta(\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-1}^{(p)}) \tag{7-50}$$

式中：L_{k-1}为$k-1$时刻存活目标的粒子数。

在预测阶段，粒子通过建议分布$q_k(. \mid \boldsymbol{x}_{k-1}, z_{1:k})$和$p(. \mid z_{1:k})$产生，即

$$\boldsymbol{x}_{k|k-1}^{(p)} \sim \begin{cases} q_k(. \mid \boldsymbol{x}_{k-1}, z_{1:k}), & p = 1, 2, \cdots, L_{k-1} \\ p(. \mid z_{1:k}), & p = L_{k-1}+1, L_{k-1}+2, \cdots, L_{k-1}+J_k \end{cases} \tag{7-51}$$

式中：J_k表示新生目标的粒子数。于是预测权值可通过下式计算为

$$w_{k|k-1}^{(p)} = \begin{cases} u(\boldsymbol{x}_{k|k-1}^{(p)}) / q_k(\boldsymbol{x}_{k|k-1}^{(p)} \mid \boldsymbol{x}_{k-1}, z_{1:k}) & p = 1, 2, \cdots, L_{k-1} \\ \gamma_k(\boldsymbol{x}_{k|k-1}^{(p)}) / p_k(\boldsymbol{x}_{k|k-1}^{(p)} \mid z_{1:k}) & p = L_{k-1}+1, \cdots, L_{k-1}+J_k \end{cases} \tag{7-52}$$

式中

$$u(\boldsymbol{x}_{k|k-1}^{(p)}) = e_{k|k-1}(\boldsymbol{x}_{k|k-1}^{(p)}) f_k(\boldsymbol{x}_{k|k-1}^{(p)} \mid \boldsymbol{x}_{k-1}^{(p)}) + b_{k|k-1}(\boldsymbol{x}_{k|k-1}^{(p)} \mid \boldsymbol{x}_{k-1}^{(p)})$$

权值更新可通过式（7-48）来完成，注意到对于$\boldsymbol{z}_k$而言，在计算似然比$l(z \mid \boldsymbol{x}_k)$时，仅那些位于$\boldsymbol{x}_{k|k-1}^{(p)}$附近的网格单元才受其影响。在这种情况下，权值的更新算式可简化为

$$w_k^{(p)} = \frac{\prod_{i \in C_i\left(\boldsymbol{x}_{k|k-1}^{(p)}\right)} \prod_{j \in C_j\left(\boldsymbol{x}_{k|k-1}^{(p)}\right)} l(z_k^{(i,j)} \mid \boldsymbol{x}_{k|k-1}^{(p)})}{\lambda_k + \psi(z_k^{(r_0,s_0)})} \tag{7-53}$$

式中

$$\psi(z_k^{(r_0,s_0)}) = \sum_{l=1}^{L_{k-1}+J_k} w_{k|k-1}^{(l)} \prod_{r \in C_r\left(\boldsymbol{x}_{k|k-1}^{(p)}\right)} \prod_{s \in C_s\left(\boldsymbol{x}_{k|k-1}^{(p)}\right)} l(z_k^{(r,s)} \mid \boldsymbol{x}_{k|k-1}^{(l)})$$

$$(r_0,s_0)=\left(D_r(\boldsymbol{x}_{k|k-1}^{(p)}),D_s(\boldsymbol{x}_{k|k-1}^{(p)})\right)$$

由此得到k时刻的更新后验密度为

$$D_k(\boldsymbol{x}_k \mid z_{1:k})=\sum_{p=1}^{L_k} w_k^{(p)}\delta(\boldsymbol{x}_k-\boldsymbol{x}_k^{(p)}) \tag{7-54}$$

在此基础上，对权值求和可得到目标数目的估计，再利用目标状态提取方法（如 k-mean 聚类算法）即可得到目标状态的估计。但从算法的实现过程看，在粒子数目较大的情况下，其运算复杂度是很大的，这是因为式（7-53）中每个粒子的权重都要逐点计算。不过，从 SMC-PHD 滤波的过程看，不难推知粒子主要集中于真实目标所处的位置附近。这就意味着，由粒子导致的有效量测数（源于目标）不会太多，这是因为大多数用于表征同一目标状态的粒子将会占据近似相同的网格单元。在这种假设条件下，可以得到一种高效的实现方法，这种方法将所有的粒子分割为P个子集，且最大的子集数目$P_{\max}=n_x\times n_y$。对于子集中的每个元素（粒子），由于具有相同的权值更新公式，因此不必反复计算，这一过程可用以下几个步骤来完成。

步骤 1：目标位置抽取

$$xPos(p)=\boldsymbol{x}_{k|k-1}^{(p)}[1], yPos(p)=\boldsymbol{x}_{k|k-1}^{(p)}[3]$$

步骤 2：位置信息离散化

$$iPos=round(xPos), jPos=round(yPos)$$

步骤 3：粒子集分割

$$SubSet[i,j]=\{\boldsymbol{x}_{k|k-1}^{(p)}:iPos(p)=i, jPos(p)=j\}$$

经上述处理后，对于某一特定的包含P_0个元素的子集$SubSet[i_0,j_0]$（该子集中的具体元素为$\{\boldsymbol{x}_{k|k-1}^{(p_l)}\}_{l=1}^{P_0}(p_l\in\{1,2,\cdots,L_{k-1}+J_k\})$），仅需进行 1 次而不是$P_0$次权值更新运算，从而可大大提高计算效率。

下面将以平面中的多目标为例，验证 SMC-PHD 的跟踪效果。仿真中假定目标状态的演化方程为

$$\boldsymbol{x}_k^s=\boldsymbol{F}\boldsymbol{x}_{k-1}^s+\boldsymbol{w}_k \quad s=1,2,\cdots,N_k \tag{7-55}$$

式中：$\boldsymbol{F}$为状态转移矩阵；$\boldsymbol{w}_k$为零均值高斯过程噪声且协方差为$\boldsymbol{Q}$。

$$\boldsymbol{F}=\mathrm{diag}[F_1,F_{1,1}]\text{，}\quad \boldsymbol{Q}=\mathrm{diag}[\boldsymbol{Q}_1,\boldsymbol{Q}_1,q_2T]$$

$$\boldsymbol{F}_1=\begin{bmatrix}1 & T\\ 0 & 1\end{bmatrix}\text{，}\quad \boldsymbol{Q}_1=q_1\cdot\begin{bmatrix}T^3/3 & T^2/2\\ T^2/2 & T\end{bmatrix}$$

式中：$q_1=0.0001$，$q_2=0.01$为目标强度过程噪声方差。

假设$T=1\mathrm{s}$，$\Delta_x=\Delta_y=1$，$n_x=n_y=20$，$\Sigma=0.7$，$\sigma=1$，利用这些参数设置产生 25 帧图像数据。初始时刻，监视区域内没有目标，在第 6 秒和第 11 秒分别出现一新目标，而在第 21 秒时第 1 个目标消失，另一个目标一直存在直到第 25 秒。在整个过程中，暂不考虑衍生目标。

目标 1 和目标 2 的初始状态分别为[4.2 0.1 7.2 0.1 *I*]和[0.0 0.1 8.2 0.12 *I*]，SMC-PHD 算法中的参数设置见表 7.2，目标的初始强度估计在 *I*–5 到 *I*+5 之间均匀采样。

表 7.2　SMC-PHD 滤波算法参数设置

变量名	γ_k	e_k	b_k	P_d	J_k	L_k	q
参数值	0.01	0.98	0	1	2000	6400	3

假设目标的初始强度分别为 I=30,20,15,10，对应的信噪比 SNR 分别为 19.8、16.3、13.8 和 10.2dB，其计算公式为 $SNR = 20\lg[I\Delta_x\Delta_y/(2\pi\Sigma^2\sigma)]$。图 7.19 给出了目标的真实航迹，图 7.20 为不同强度下得到的第 15 帧图像。

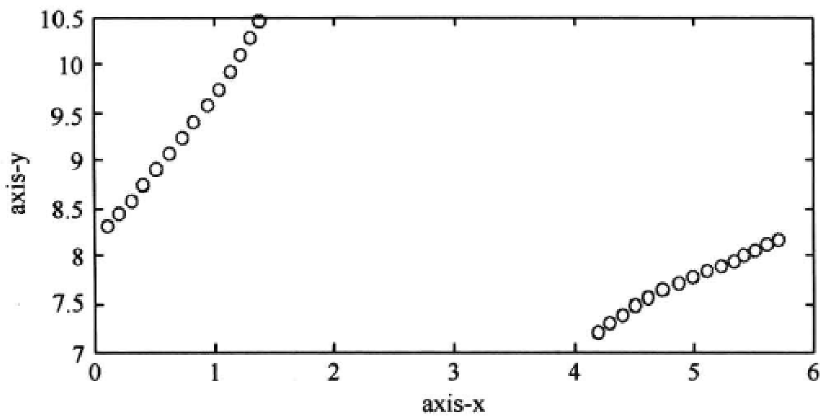


图 7.19　目标航迹示例

(a) I=30

(b) I=20

(c) I=15

(d) I=10

图 7.20　目标强度图像

由图 7.20 中的结果可以看出，背景噪声中的目标是较为微弱的，即便是在I=30 的情况下，仍存在大量的虚警，此时采用传统先检测后跟踪的方法很可能丢失目标，很难得到理想的效果。为此这里主要考虑基于 SMC-PHD 的检测前跟踪（TbD）方法。

进一步假设目标初始强度I=30，采用 SMC-PHD 算法进行跟踪。对于多目标跟踪，不仅要估计目标数目，还要估计各目标的状态，此时单目标应用条件下的性能测度方法已不能满足要求。为此，这里采用最优子模式分配（Optimal Sub-Pattern Assignment，OSPA）[28]距离指标进行测度，其定义为

$$\bar{d}_p^{(c)}(X,\hat{X})=\left[\frac{1}{n}\left(\min_{\pi\in\Pi_k}\sum_{i=1}^{m}d_p^{(c)}\left(\boldsymbol{x}_i,\hat{\boldsymbol{x}}_{\pi(i)}\right)^p+c_p(n-m)\right)\right]^{1/p} \tag{7-56}$$

式中：X 和 $\hat{X}$ 为任意子集，维数分别为 m 和 n，且 $m\leqslant n$；$1\leqslant p<\infty$，$d^{(c)}(\boldsymbol{x},\hat{\boldsymbol{x}})=\min\left(c,d(\boldsymbol{x},\hat{\boldsymbol{x}})\right)$，$c>0$，$d(\boldsymbol{x},\hat{\boldsymbol{x}})=\|\boldsymbol{x}-\hat{\boldsymbol{x}}\|$；$\Pi_k$ 表示$\{1,2,\cdots,k\}$的所有排列组成的集合。若 $m>n$，则可令 $\bar{d}_p^{(c)}(X,\hat{X})=\bar{d}_p^{(c)}(\hat{X},X)$。

对于多次仿真，则需采用 OSPA 的均方根形式，其定义为

$$d_{rms}(k)=\left[\frac{1}{M}\sum_{i=1}^{M}\left(\bar{d}_p^{(c)}(i,k)\right)^2\right]^{1/2} \tag{7-57}$$

式中：M 为实验次数；$\bar{d}_p^{(c)}(i,k)$ 为第 i 次实验的 OPSA 距离。

采用随机产生粒子进行初始化的方法进行跟踪，即目标位置在监视区域[0 20]×[0 20] 内均匀取值，而目标速度则在[0 0.2]×[0 0.2] 范围内均匀取值，总的粒子数目 N=10000，其中用于新生目标检测的粒子数目为 2000。取c=20，p=2，仿真次数为 M=100，所得结果如图 7.21～图 7.23 所示。

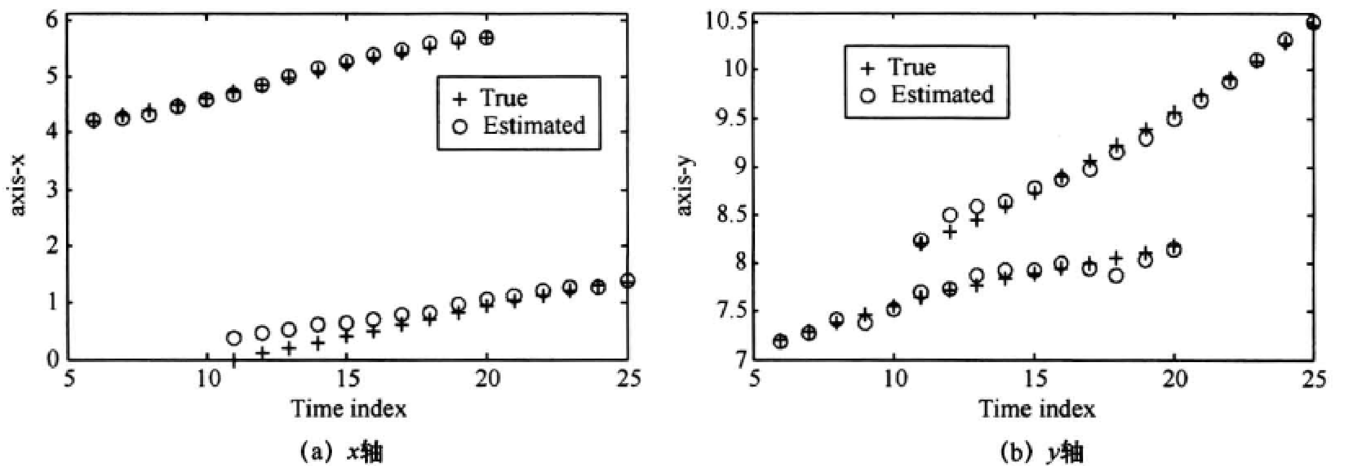


图 7.21 目标位置估计结果

由图 7.21～图 7.23 中的结果可以看出，采用 SMC-PHD 跟踪算法，可实现对可变数目条件下的多目标检测前跟踪，且对目标数目的估计较为准确、可靠。从图 7.23 的结果看，当目标数目发生变化的时（尤其是在有新生目标出现时），OSPA 距离会出现增大的现象，这与目标数目发生变化时将引入新的误差的事实是一致的。随着跟踪过程的持续，目标状态的估计逐渐趋于稳定，相应的 OSPA 距离也随之降低。

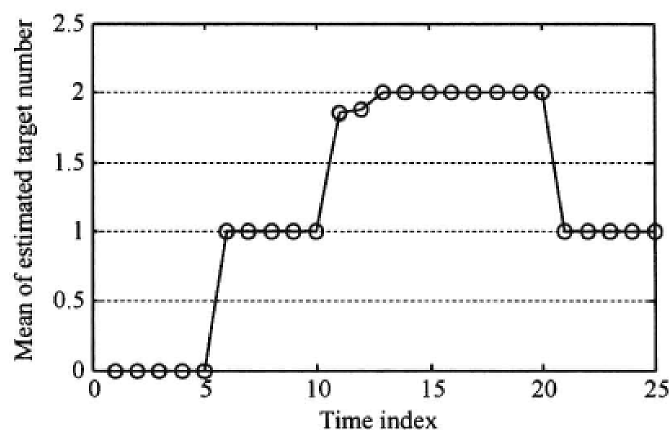


图 7.22 目标数目估计均值

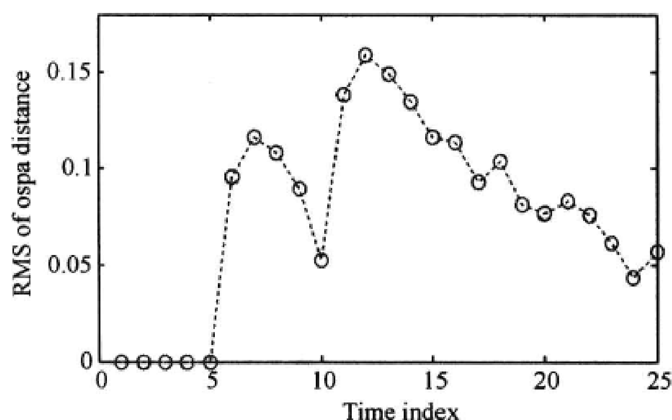


图 7.23 OSPA 距离均方根

需要指出的是，本实例中考虑的是在无任何先验信息条件下的粒子初始化方法（在整个监视区域随机均匀分布）；这种通用设置能适应复杂的跟踪条件，但相应地粒子的效率也较低，同时也影响了跟踪性能的提高。另外，本实例中考虑的目标场景相对较为简单，对于具有多个临近目标或目标存在交叉航迹等复杂情况，需要进行一些特殊的处理，感兴趣的读者可参阅相关文献如文献[29]等。

7.6 本章小结

本章主要阐述基于非参数化蒙特卡罗仿真技术来实现递推贝叶斯滤波的原理和实现方法，首先从贝叶斯滤波问题中多维积分式的求解入手，阐述了随机采样近似原理和基于重要采样技术的序贯处理方法；在此基础上针对采样函数为系统先验（状态转移）函数的一类特殊情况，重点阐述了标准粒子滤波（SPF）算法的具体处理流程，并给出了制导雷达跟踪和多目标检测前跟踪的应用实例。

实验结果表明，与传统的解析高斯近似方法相比，以随机采样近似为基础的粒子滤波算法由于具有粒子散布性的特点，在提高跟踪系统的稳定性和收敛速度方面具有明显的优势。另外，作为一种通用的贝叶斯推演手段，粒子滤波不仅能够处理传统高斯型滤波方法所能处理的非线性估计问题，还能处理非高斯、多模态条件下的复杂问题，具有更广泛的适应性和更巨大的应用潜能。

需要指出的是，粒子滤波的优势在于其通用性，这种优势不是针对任何非线性滤波问题都具备的，这是因为：①作为一种基于随机采样的实现方法，粒子滤波算法必须以大量的粒子为基础，运算开销较大；②对于某些简单条件下的非线性滤波问题，如基于目标距离、方位量测的跟踪，传统的解析高斯近似方法本身已能达到较理想的效果，采用粒子滤波手段所能获得的额外效果甚微。因此，在实际应用中，应该根据待处理问题的特点对拟采用的方法进行合理选择。

第 8 章　优选分布采样滤波

从第 7 章的分析知道，SPF 直接根据状态转移方程（或先验分布）产生新的粒子，由于没有考虑新近观测对状态估计的影响（即粒子的采样完全脱离了观测信息的指导），这种滤波器在很多情况下表现出较差的性能，即便在粒子数目很大的情况下也是如此。

为了提高粒子滤波算法的估计性能，各种基于重要性函数采样的粒子滤波方法应运而生，这类方法的共同点是在产生粒子时融入了观测信息，因而所得到的粒子更加有效；不同点在于指导粒子产生的方法存在差异，即重要采样函数（或建议分布）的选择不同。理论上，为了最大限度地提高粒子的质量和效能，希望得到用于指导粒子生成的最优采样函数。但是对于一般的非线性系统，最优重要采样函数很难求得或根本不存在，只能通过近似的方法来获取次优的建议分布。

本章主要研究和分析几种基于重要采样函数的粒子滤波算法，包括最优采样函数近似滤波粒子算法、EKF 近似粒子算法、UPF 及其修正算法等，针对粒子退化问题，介绍了用于保持粒子多样性的马尔可夫链蒙特卡罗（MCMC）重采样方法，并结合目标跟踪应用给出了两个典型的实例。

8.1　最优函数近似采样滤波

8.1.1　最优重要函数近似

在前面的讨论中提到，选择合适的重要函数可以有效减小权值的方差，从而减轻粒子退化问题。通常情况下，对于一般的非线性系统

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k) + \boldsymbol{w}_k \tag{8-1}$$

$$\boldsymbol{y}_k = h(\boldsymbol{x}_k) + \boldsymbol{v}_k \tag{8-2}$$

很难得到最优重要函数 $p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}, \boldsymbol{y}_k)$ 的解析表达解，但若系统方程满足如下形式

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k) + \boldsymbol{w}_k \qquad \boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_k) \tag{8-3}$$

$$\boldsymbol{y}_k = \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{v}_k \qquad \boldsymbol{v}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_k) \tag{8-4}$$

可以证明（见附录 E），最优采样函数的解析表达式为

$$p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}, \boldsymbol{y}_k) = \mathcal{N}(\boldsymbol{x}_k; \boldsymbol{m}_k, \boldsymbol{P}_k) \tag{8-5}$$

式中

$$\boldsymbol{m}_k = \boldsymbol{P}_k(\boldsymbol{Q}_k^{-1} f(\boldsymbol{x}_{k-1}) + \boldsymbol{H}_k^{\mathrm{T}} \boldsymbol{R}_k^{-1} \boldsymbol{y}_k), \quad \boldsymbol{P}_k^{-1} = \boldsymbol{Q}_k^{-1} + \boldsymbol{H}_k^{\mathrm{T}} \boldsymbol{R}_k^{-1} \boldsymbol{H}_k$$

对于式（8-2）所表示的一般非线性观测，虽然无法直接求得最优解，但可通过对模型进行局部线性化处理，求得近似最优解。特别地，若对式（8-2）进行一阶泰勒展开，得

$$\boldsymbol{y}_k \approx h(f(\boldsymbol{x}_{k-1})) + \boldsymbol{H}_k[\boldsymbol{x}_k - f(\boldsymbol{x}_{k-1})] + \boldsymbol{v}_k \tag{8-6}$$

式中

$$\boldsymbol{H}_k = \nabla_x h(\boldsymbol{x})\big|_{x=f(x_{k-1})}$$

对照式（8-4），易知此时有

$$\hat{\boldsymbol{m}}_k = \boldsymbol{P}_k\{\boldsymbol{Q}_k^{-1} f(\boldsymbol{x}_{k-1}) + \boldsymbol{H}_k^{\mathrm{T}} \boldsymbol{R}_k^{-1}[\boldsymbol{y}_k - h(f(\boldsymbol{x}_{k-1})) + \boldsymbol{H}_k f(\boldsymbol{x}_{k-1})]\} \tag{8-7}$$

$$\hat{\boldsymbol{P}}_k = (\boldsymbol{Q}_k^{-1} + \boldsymbol{H}_k^{\mathrm{T}} \boldsymbol{R}_k^{-1} \boldsymbol{H}_k)^{-1} \tag{8-8}$$

即近似最优采样函数为 $\hat{p}(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}, \boldsymbol{y}_k) = \mathcal{N}(\boldsymbol{x}_k; \hat{\boldsymbol{m}}_k, \hat{\boldsymbol{P}}_k)$，于是可通过该采样函数引导粒子的生成。

8.1.2 MCMC 重采样

由 Alg（7.1）的（3）可知，重采样是根据计算所得的权重对粒子或舍弃或多份复制，这一过程引入的一个负面影响是当大多数粒子的权重较小而只有少数（极端情况下只有 1 个）粒子占有绝对性的权重时，经重采样后将得到大量完全相同的粒子，粒子的多样性丧失，导致粒子贫化问题。显然，在这种情况下，极其少量的有效粒子很难对真实的状态分布进行准确的描述，算法性能将会严重恶化。文献[30]指出，通过某种马尔可夫链进行蒙特卡罗积分运算，可以保持状态分布不变。为此，可以借助这一思想，在重采样之后引入 MCMC 步骤，使粒子向所期望的状态空间移动，在恢复粒子多样性的同时不改变原有的分布。其基本思路为：在获得重采样的粒子 $\tilde{\boldsymbol{x}}_k^{(j)}$ 后，通过马尔可夫链转移核 $\psi(\boldsymbol{x}_k^{(j)} \mid \tilde{\boldsymbol{x}}_k^{(j)})$ 产生一组新的粒子 $\boldsymbol{x}_k^{(j)}$ 使之满足

$$\int_{\tilde{\boldsymbol{x}}_k^{(j)}} \psi(\boldsymbol{x}_k^{(j)} \mid \tilde{\boldsymbol{x}}_k^{(j)}) p(\tilde{\boldsymbol{x}}_k^{(j)} \mid \tilde{\boldsymbol{x}}_{0:k-1}^{(j)}, \boldsymbol{y}_{1:k}) \mathrm{d}\tilde{\boldsymbol{x}}_k^{(j)} = p(\boldsymbol{x}_k^{(j)} \mid \tilde{\boldsymbol{x}}_{0:k-1}^{(j)}, \boldsymbol{y}_{1:k}) \tag{8-9}$$

通常，上述不变性条件可通过 Metropolis-Hastings 算法来实现，其基本流程可概括如下。

Alg（8.1）：Metropolis-Hastings Algorithm

（1）根据当前时刻 k 得到的重采样索引 j，用 $\boldsymbol{x}_{k-1}^{(j)}$ 取代 $\boldsymbol{x}_{k-1}^{(i)}$。

（2）对每一个粒子 $\boldsymbol{x}_{k-1}^{(j)}$，通过式（8-7）和式（8-8）计算 $\hat{\boldsymbol{m}}_k^{(j)}$ 和 $\hat{\boldsymbol{P}}_k^{(j)}$。

（3）对 IDF 进行采样：$\boldsymbol{x}_k^{*(j)} \sim \mathcal{N}(\boldsymbol{x}_k; \hat{\boldsymbol{m}}_k^{(j)}, \hat{\boldsymbol{P}}_k^{(j)})$。

（4）产生满足均匀分布的随机数 $a \sim \mathcal{U}(0,1)$。

（5）拒绝/接受。

若 $a \leqslant \min\{1, w(\boldsymbol{x}_k^{*(j)}) / w(\tilde{\boldsymbol{x}}_k^{(j)})\}$，$\boldsymbol{x}_k^{(j)} = \boldsymbol{x}_k^{*(j)}$；否则，$\boldsymbol{x}_k^{(j)} = \tilde{\boldsymbol{x}}_k^{(j)}$。

8.1.3 滤波算法流程

综合以上分析，改进的粒子滤波算法（这里称为 IPF）完整的计算流程如下[31，32]。

Alg（8.2）：Improved Particle Filtering（IPF）

（1）由先验分布 $p(\boldsymbol{x}_0)$ 得到 N 个采样 $\{\boldsymbol{x}_0^{(i)}\}_{i=1}^N$。

（2）根据式（8-7）和式（8-8）计算 $\hat{\boldsymbol{m}}_k^{(i)}$、$\hat{\boldsymbol{P}}_k^{(i)}$ 得到粒子 $\hat{\boldsymbol{x}}_k^{(i)} \sim \mathcal{N}(\boldsymbol{x}_k; \boldsymbol{m}_k^{(i)}, \hat{\boldsymbol{P}}_k^{(i)})$。

（3）计算 $w_k^{(i)} \propto w_{k-1}^{(i)} \dfrac{p(\boldsymbol{y}_k \mid \hat{\boldsymbol{x}}_k^{(i)}) p(\hat{\boldsymbol{x}}_k^{(i)} \mid \boldsymbol{x}_{k-1}^{(i)})}{\mathcal{N}(\hat{\boldsymbol{x}}_k^{(i)} - \boldsymbol{m}_k^{(i)}, \hat{\boldsymbol{P}}_k^{(i)})}$ 并归一化得 $\tilde{w}_k^{(i)}$。

（4）对粒子 $\{\hat{\boldsymbol{x}}_k^{(i)}\}_{i=1}^N$ 进行重采样得到一组新粒子 $\{\tilde{\boldsymbol{x}}_k^{(j)}\}_{j=1}^N$，使其满足 $\Pr(\tilde{\boldsymbol{x}}_k^{(j)}=\hat{\boldsymbol{x}}_k^{(i)})=\tilde{w}_k^{(i)}$。

（5）MCMC 步骤。

（6）重复步骤（2）～步骤（5）。

与标准粒子滤波（SPF）相比，改进的算法（IPF）在采样函数中融入了新近的量测，能使产生的粒子更好地近似真实的状态分布；同时，MCMC 过程能有效防止粒子贫化现象的发生，从而提高算法的性能。

8.2　EKF 近似粒子滤波

在 8.1 节中，围绕最优采样函数 $p(\boldsymbol{x}_k\mid\boldsymbol{x}_{k-1},\boldsymbol{y}_k)$ 的近似问题，介绍了线性化展开近似的处理方法。形式上，$p(\boldsymbol{x}_k\mid\boldsymbol{x}_{k-1},\boldsymbol{y}_k)$ 表达的是利用 $k-1$ 时刻状态 $\boldsymbol{x}_{k-1}$ 和 k 时刻量测信息 $\boldsymbol{y}_k$ 获取 k 时刻 $\boldsymbol{x}_k$ 的最优估计问题，通常情况下 $p(\boldsymbol{x}_k\mid\boldsymbol{x}_{k-1},\boldsymbol{y}_k)$ 的解析表达式并不存在，实际应用中，为了处理问题的方便，常将其近似为高斯分布。由于 EKF 提供了近似求解 $p(\boldsymbol{x}_k\mid\boldsymbol{x}_{k-1},\boldsymbol{y}_k)$ 的一条重要的途径，自然地可将其与传统的粒子滤波结合得到 EPF（Extended Particle Filtering）算法。接下来将详细介绍该算法的原理和实现方法。

在 EPF 算法中，通过 $k-1$ 时刻的 N 个粒子 $\{\boldsymbol{x}_{k-1}^{(i)},w_{k-1}^{(i)}\}_{i=1}^N$ 利用 EKF 更新方法得到 k 时刻的 N 个粒子集 $\{\boldsymbol{x}_k^{(j)},w_k^{(j)}\}_{j=1}^N$，对每一个粒子 $\boldsymbol{x}_{k-1}^{(i)}$，上述过程可表示为

$$p(\boldsymbol{x}_k\mid\boldsymbol{x}_{k-1},\boldsymbol{y}_k)\approx\pi_{\mathcal{N}}(\boldsymbol{x}_k\mid\boldsymbol{y}_{1:k})=\mathcal{N}(\boldsymbol{x}_k,\boldsymbol{x}_k^{(i)},\boldsymbol{P}_k^{(i)})$$

实际处理时，可采用 N 个并行的 EKF 滤波器组来实现，具体算法流程可概括如下。

Alg（8.3）：Extended Particle Filtering　（EPF）

步骤 1：初始化：$k=0$

从先验状态分布 $p(\boldsymbol{x}_0)=\mathcal{N}(\boldsymbol{x}_0;\hat{x}_0^{(i)},\boldsymbol{P}_0^{(i)})$ 抽取一组粒子 $\{\boldsymbol{x}_0^{(i)}\}_{i=1}^N$。

For　$k=1,2,\cdots$

步骤 2：重要采样过程

For　$i=1,2,\cdots,N$

（1）用 EKF 对每一粒子进行状态更新。

时间更新：

$$\hat{\boldsymbol{x}}_{k|k-1}^{(i)}=f(\hat{\boldsymbol{x}}_{k-1}^{(i)}) \tag{8-10}$$

$$\boldsymbol{P}_{k|k-1}^{(i)}=\boldsymbol{F}_k^{(i)}\boldsymbol{P}_{k-1}^{(i)}(\boldsymbol{F}_k^{(i)})^{\mathrm{T}}+\boldsymbol{Q}_k \tag{8-11}$$

式中：$\boldsymbol{F}_k^{(i)}=\nabla_{\boldsymbol{x}}f(\boldsymbol{x})\big|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{k-1}^{(i)}}$。

测量更新：

$$\boldsymbol{K}_k^{(i)}=\boldsymbol{P}_{k|k-1}^{(i)}(\boldsymbol{H}_k^{(i)})^{\mathrm{T}}[\boldsymbol{H}_k^{(i)}\boldsymbol{P}_{k|k-1}^{(i)}(\boldsymbol{H}_k^{(i)})^{\mathrm{T}}+\boldsymbol{R}_k]^{-1} \tag{8-12}$$

$$\hat{\boldsymbol{x}}_k^{(i)}=\hat{\boldsymbol{x}}_{k|k-1}^{(i)}+\boldsymbol{K}_k(\boldsymbol{y}_k-\hat{\boldsymbol{y}}_{k|k-1}^{(i)}) \tag{8-13}$$

$$\boldsymbol{P}_k^{(i)}=\boldsymbol{P}_{k|k-1}^{(i)}-\boldsymbol{K}_k^{(i)}\boldsymbol{H}_k^{(i)}\boldsymbol{P}_{k|k-1}^{(i)} \tag{8-14}$$

式中：$\boldsymbol{H}_k^{(i)}=\nabla_{\boldsymbol{x}}h(\boldsymbol{x})\big|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{k|k-1}^{(i)}}$。

（2）从 IDF 进行采样。

$$x_k^{(i)} \sim \pi_{\mathcal{N}}(x_k \mid y_{1:k}) = \mathcal{N}(x_k; \hat{x}_k^{(i)}, P_k^{(i)}) \tag{8-15}$$

For $i = 1, 2, \cdots, N$，估算每一粒子的权重并进行归一化：

$$w_k^{(i)} = w_{k-1}^{(i)} p(y_k \mid x_k^{(i)}) p(x_k^{(i)} \mid x_{k-1}^{(i)}) / \pi_{\mathcal{N}}(x_k^{(i)} \mid y_{1:k}) \tag{8-16}$$

$$\tilde{w}_k^{(i)} = w_k^{(i)} / \sum_{i=1}^{N} w_k^{(i)} \tag{8-17}$$

步骤 3：重采样过程

对粒子 $\{x_k^{(i)}\}_{i=1}^N$ 进行采样（选择）得到 N 个新的样本 $\{x_k^{(l)}\}_{l=1}^N$，并使其概率满足

$$\Pr(x_k^{(l)} = x_k^{(i)}) = \tilde{w}_k^{(i)} \tag{8-18}$$

步骤 4：估计输出

$$\hat{x}_k = E[x_k \mid y_{1:k}] \approx \sum_{i=1}^{N} \tilde{w}_k^{(i)} x_k^{(i)} \tag{8-19}$$

通过执行步骤 1～步骤 3 的递推步骤，可以序贯得到状态量的估计。

8.3 UPF 算法及其修正

在前面的分析中指出，粒子滤波的关键是如何产生有效的采样来近似后验 PDF，实质上就是寻求便于实现且又能较好近似真实分布的 IDF。鉴于 UKF 算法本身能够有效融合观测信息，并将粒子推向高似然的区域（由新近的观测量 y_k 来实现），从而对真实的 PDF 进行有效近似，文献[24]中提出了用 UKF 的滤波结果作为建议分布的思想。由于该方法比直接用先验或由 EKF 产生建议分布的算法性能都好，因此在很多应用场合受到重视[33-35]。但是，在实际应用中发现，UPF 同样存在潜在的不足，特别是在过程噪声协方差 Q_k 与估计误差协方差 P_k 相差悬殊的情况下极易出现数值敏感问题，并由此导致严重的粒子贫化现象。在许多应用特别是被动目标跟踪应用中，由于跟踪误差一般都较大，而过程噪声通常要小得多，因此 Q_k 与 P_k 相差悬殊是一种较为普遍的现象。有鉴于此，本节将在传统 UPF 的基础上提出一种修正算法（MUPF），该算法通过修正权值的计算避免了数值敏感问题，并改善了权值的分布特性（即降低权方差），即便在 Q_k 奇异的情况下也能进行方便的计算。

在接下来的几节中，首先将对传统 UPF 算法作一简要回顾，在此基础上讨论数值敏感问题如何影响滤波器的性能，同时提出一种修正算法来有效解决 UPF 中存在的不足。

8.3.1 UPF 算法

UPF 算法的基本思想是采用 UKF 为每一粒子产生所需的 IDF $\pi_{\mathcal{N}}(x_k \mid y_{1:k})$，以此来近似最优采样函数，即

$$p(x_k \mid x_{k-1}, y_k) \approx \pi_{\mathcal{N}}(x_k \mid y_{1:k}) \tag{8-20}$$

通过对这一分布函数进行采样可得到新的粒子。关于 UPF 算法的详细推导可参阅文献[24]，这里仅给出算法的计算流程。

Alg（8.4）：Unscented Particle Filtering （UPF）

步骤 1：初始化：$k = 0$

从先验状态分布 $p(\boldsymbol{x}_0)=\mathcal{N}(\boldsymbol{x}_0;\hat{\boldsymbol{x}}_0,\boldsymbol{P}_0)$ 抽取一组粒子 $\{\boldsymbol{x}_0^{(i)}\}_{i=1}^N$，同时定义一组参数：$c_0=\kappa/(L+\kappa)$，$c_j=c_{j+L}=0.5/(L+\kappa)$，$\gamma=\sqrt{L+\kappa}$。其中 $j=1,2,\cdots,L$，L 为状态 $\boldsymbol{x}$ 的维数；γ 为尺度因子。

$$\text{For}\quad k=1,2,\cdots$$

步骤 2：重要采样过程

$$\text{For}\quad i=1,2,\cdots,N$$

（1）用 UKF 对每一粒子进行状态更新。

产生变换样点：

$$\chi_{k-1,0:2L}^{(i)}=[\hat{\boldsymbol{x}}_{k-1}^{(i)}\quad \hat{\boldsymbol{x}}_{k-1}^{(i)}+\gamma\sqrt{\boldsymbol{P}_{k-1}^{(i)}}\quad \hat{\boldsymbol{x}}_{k-1}^{(i)}-\gamma\sqrt{\boldsymbol{P}_{k-1}^{(i)}}] \tag{8-21}$$

时间更新：

$$\chi_{k|k-1,0:2L}^{(i)}=f(\chi_{k-1,0:2L}^{(i)}) \tag{8-22}$$

$$\hat{\boldsymbol{x}}_{k|k-1}^{(i)}=\sum_{j=0}^{2L}c_j\chi_{k|k-1,j}^{(i)} \tag{8-23}$$

$$\boldsymbol{P}_{k|k-1}^{(i)}=\sum_{j=0}^{2L}c_j(\chi_{k|k-1,j}^{(i)}-\hat{\boldsymbol{x}}_{k|k-1}^{(i)})(\chi_{k|k-1,j}^{(i)}-\hat{\boldsymbol{x}}_{k|k-1}^{(i)})^{\mathrm{T}}+\boldsymbol{Q}_k \tag{8-24}$$

$$\chi_{k|k-1,0:2L_a}^{*(i)}=\left[\chi_{k|k-1,0:2L}^{(i)}\quad \chi_{k|k-1,0}^{(i)}+\gamma\sqrt{\boldsymbol{Q}_k}\quad \chi_{k|k-1,0}^{(i)}-\gamma\sqrt{\boldsymbol{Q}_k}\right] \tag{8-25}$$

$$\mathcal{Y}_{k|k-1,0:2L_a}^{*(i)}=h(\chi_{k|k-1,0:2L_a}^{*(i)}) \tag{8-26}$$

$$\hat{\boldsymbol{y}}_{k|k-1}^{(i)}=\sum_{j=0}^{2L_a}c_j^*\mathcal{Y}_{k|k-1,j}^{(i)} \tag{8-27}$$

测量更新：

$$\boldsymbol{P}_{\boldsymbol{y}_k\boldsymbol{y}_k}=\sum_{j=0}^{2L_a}c_j^*(\mathcal{Y}_{k|k-1,j}^{*(i)}-\hat{\boldsymbol{y}}_{k|k-1}^{(i)})(\mathcal{Y}_{k|k-1,j}^{*(i)}-\hat{\boldsymbol{y}}_{k|k-1}^{(i)})^{\mathrm{T}}+\boldsymbol{R}_k \tag{8-28}$$

$$\boldsymbol{P}_{\boldsymbol{x}_k\boldsymbol{y}_k}=\sum_{j=0}^{2L_a}c_j^*(\chi_{k|k-1,j}^{*(i)}-\hat{\boldsymbol{x}}_{k|k-1}^{(i)})(\mathcal{Y}_{k|k-1,j}^{*(i)}-\hat{\boldsymbol{y}}_{k|k-1}^{(i)})^{\mathrm{T}} \tag{8-29}$$

$$\boldsymbol{K}_k=\boldsymbol{P}_{\boldsymbol{x}_k\boldsymbol{y}_k}\boldsymbol{P}_{\boldsymbol{y}_k\boldsymbol{y}_k}^{-1} \tag{8-30}$$

$$\hat{\boldsymbol{x}}_k^{(i)}=\hat{\boldsymbol{x}}_{k|k-1}^{(i)}+\boldsymbol{K}_k(\boldsymbol{y}_k-\hat{\boldsymbol{y}}_{k|k-1}^{(i)}) \tag{8-31}$$

$$\boldsymbol{P}_k^{(i)}=\boldsymbol{P}_{k|k-1}^{(i)}-\boldsymbol{K}_k\boldsymbol{P}_{\boldsymbol{y}_k\boldsymbol{y}_k}\boldsymbol{K}_k^{\mathrm{T}} \tag{8-32}$$

式中

$$L_a=2L\,,\quad c_0^*=\kappa/(L_a+\kappa)\,,\quad c_j^*=c_{j+L_a}^*=0.5/(L_a+\kappa)$$

（2）从 IDF 进行采样。

$$\boldsymbol{x}_k^{(i)}\sim\pi_{\mathcal{N}}(\boldsymbol{x}_k\mid\boldsymbol{y}_{1:k})=\mathcal{N}(\boldsymbol{x}_k;\hat{\boldsymbol{x}}_k^{(i)},\boldsymbol{P}_k^{(i)}) \tag{8-33}$$

For $i=1,2,\cdots,N$

估算每一粒子的权重并进行归一化：

$$w_k^{(i)}=w_{k-1}^{(i)}p(\boldsymbol{y}_k\mid\boldsymbol{x}_k^{(i)})p(\boldsymbol{x}_k^{(i)}\mid\boldsymbol{x}_{k-1}^{(i)})/\pi_{\mathcal{N}}(\boldsymbol{x}_k^{(i)}\mid\boldsymbol{y}_{1:k}) \tag{8-34}$$

$$\tilde{w}_k^{(i)}=w_k^{(i)}/\sum\nolimits_{i=1}^N w_k^{(i)} \tag{8-35}$$

步骤 3：重采样过程

对粒子 $\{\boldsymbol{x}_k^{(i)}\}_{i=1}^N$ 进行采样（选择）得到 N 个新的样本 $\{\boldsymbol{x}_k^{(l)}\}_{l=1}^N$，并使其概率满足

$$\Pr(\boldsymbol{x}_k^{(l)} = \boldsymbol{x}_k^{(i)}) = \tilde{w}_k^{(i)} \tag{8-36}$$

步骤 4：估计输出

$$\hat{\boldsymbol{x}}_k = E[\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k}] \approx \sum_{i=1}^N \tilde{w}_k^{(i)} \boldsymbol{x}_k^{(i)} \tag{8-37}$$

通过执行步骤 1～步骤 3 的递推步骤，即可得到状态量的估计。

8.3.2 修正的 UPF 算法

UPF 算法（通过 UKF 生成建议分布）被认为是最有效的估计方法之一，但在实际应用中发现存在这样几个问题：①当协方差矩阵 $\boldsymbol{Q}_k$ 奇异时，式（8-34）中的 $p(\boldsymbol{x}_k^{(i)} \mid \boldsymbol{x}_{k-1}^{(i)})$ 计算较为困难；②当状态估计误差协方差矩阵 $\boldsymbol{P}_k$ 和过程噪声协方差矩阵 $\boldsymbol{Q}_k$ 在量值上不可比拟（这在实际问题中非常普遍）时，数值敏感性问题（即计算溢出）就会随之产生；③由于②，算法将会遭受严重的粒子退化问题，这是因为在每次递推过程中，绝大多数粒子的权值小到可以忽略，仅有个别的粒子占有较大的权重。下面以图 8.1 所示的简单二维状态估计问题为例，解释这种现象出现的原因。

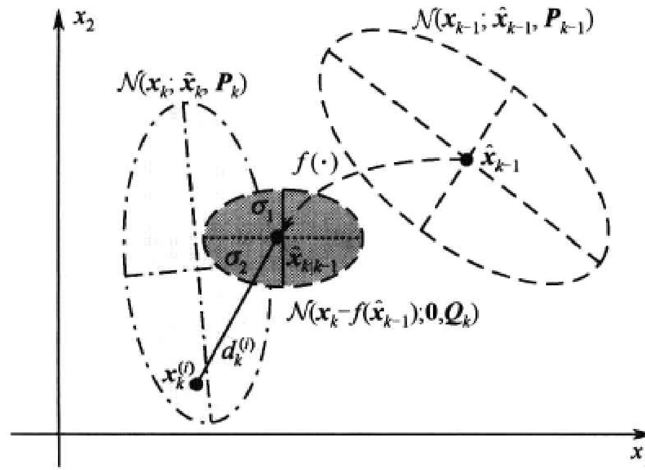


图 8.1 二维状态空间中的先验计算示例

如图 8.1 所示，在 $k-1$ 时刻，状态分布满足 $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{x}_{k-1}; \hat{\boldsymbol{x}}_{k-1}, \boldsymbol{P}_{k-1})$；当经过系统模型传递之后，其分布可记为 $p(\boldsymbol{x}_k \mid \hat{\boldsymbol{x}}_{k-1}) = \mathcal{N}(\boldsymbol{x}_k - f(\hat{\boldsymbol{x}}_{k-1}); \boldsymbol{0}, \boldsymbol{Q}_k)$。为讨论方便，这里假设矩阵 $\boldsymbol{Q}_k$ 为对角阵且 $\boldsymbol{Q}_k = \text{diag}\{\sigma_1^2, \sigma_2^2\}$。在假设条件下，显然状态预测误差 $\boldsymbol{e}_k = \boldsymbol{x}_k - f(\hat{\boldsymbol{x}}_{k-1})$ 在两轴长度分别为 σ_1 和 σ_2 的椭圆体所包含的范围之内。但是，经过测量更新并通过重要函数采样（式（8-33））之后，由于式（8-33）引入了较大的采样不确定性，$d_k^{(i)} = \|\boldsymbol{x}_k^{(i)} - f(\hat{\boldsymbol{x}}_{k-1})\|$ 和 $\sigma_{\max} = \max\{\sigma_1, \sigma_2\}$ 的量值很可能不在同一数量级。一方面，这会引起数值敏感问题；另一方面，若忽略所有的非显著性权值，则会进一步导致粒子退化问题。为详细说明这一点，首先将状态误差表示为

$$\boldsymbol{e}_k^{(i)} = \boldsymbol{x}_k^{(i)} - f(\hat{\boldsymbol{x}}_{k-1}) = [\Delta x_{1,k}^{(i)} \ \Delta x_{2,k}^{(i)}]^{\mathrm{T}} \tag{8-38}$$

进一步有

$$
\begin{aligned}
(\boldsymbol{e}_k^{(i)})^{\mathrm{T}} \boldsymbol{Q}_k^{-1} \boldsymbol{e}_k^{(i)} &= \sigma_1^{-2}(\Delta x_{1,k}^{(i)})^2 + \sigma_2^{-2}(\Delta x_{2,k}^{(i)})^2 \\
&\geqslant \sigma_{\max}^{-2}[(\Delta x_{1,k}^{(i)})^2 + (\Delta x_{2,k}^{(i)})^2] = (\sigma_{\max}^{-1} d_k^{(i)})^2
\end{aligned} \tag{8-39}
$$

由式（8-39）可以清楚地知道，若 $d_k^{(i)} \gg \sigma_{\max}$ 成立，则状态先验

$$
p(\boldsymbol{x}_k^{(i)} \mid \boldsymbol{x}_{k-1}^{(i)}) = const \cdot \exp[-0.5 \cdot (\boldsymbol{e}_k^{(i)})^{\mathrm{T}} \boldsymbol{Q}_k^{-1} \boldsymbol{e}_k^{(i)}] \to 0 \tag{8-40}
$$

事实上，这一现象是非常普遍的，这是因为通常情况下过程噪声相对较小，而状态估计误差可能会很大，相应地其协方差为了能包含估计误差也将会非常大。

通过上述分析可知，问题的根源在于式（8-34）中的重要权计算。为避免数值敏感问题，一种可能的方法是先计算对数权，而后通过忽略共同的比例因子得到归一化的权值，这种方法在某些情况下可能奏效。但是，一方面，若 $d_k^{(i)} \in [d_{\min}\ d_{\max}]$ 存在于一个很大的数值区间，数值敏感性问题仍然不能解决，也就无法得到粒子的权重；另一方面，即便在某些情况下通过计算对数权可以消除数值敏感问题，但由于权值分布的极端不均匀性，仍会不可避免地出现严重的粒子退化问题。接下来，将提出一种修正形式的权值计算方法[36]，来有效解决前述的几点不足。

实际上，在 UPF 框架下，状态的后验分布是用混合高斯模型近似的，即在 $k-1$ 时刻，有

$$
p(\boldsymbol{x}_{k-1} \mid \boldsymbol{y}_{1:k-1}) = \sum_{i=1}^{N} \tilde{w}_{k-1}^{(i)} \mathcal{N}(\boldsymbol{x}_{k-1}; \hat{\boldsymbol{x}}_{k-1}^{(i)}, \boldsymbol{P}_{k-1}^{(i)}) \tag{8-41}
$$

为简便起见，这里将以单个粒子为例，讨论其状态更新过程，其基本目标是由 $p(\boldsymbol{x}_{k-1} \mid \boldsymbol{y}_{1:k-1})$ 的第 i 个分量得到后验分布 $p(\boldsymbol{x}_k^{(i)} \mid \boldsymbol{y}_{1:k})$ 的估计。根据贝叶斯规则，预测分布可计算为

$$
\begin{aligned}
p(\boldsymbol{x}_k^{(i)} \mid \boldsymbol{y}_{1:k-1}) &= \int p(\boldsymbol{x}_k^{(i)} \mid \boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{y}_{1:k-1}) p(\boldsymbol{x}_{k-1}^{(i)} \mid \boldsymbol{y}_{1:k-1}) \mathrm{d}\boldsymbol{x}_{k-1}^{(i)} \\
&= \int \mathcal{N}(\boldsymbol{x}_k^{(i)}; f(\boldsymbol{x}_{k-1}^{(i)}), \boldsymbol{Q}_k) p(\boldsymbol{x}_{k-1}^{(i)} \mid \boldsymbol{y}_{1:k-1}) \mathrm{d}\boldsymbol{x}_{k-1}^{(i)}
\end{aligned} \tag{8-42}
$$

在 UKF 算法中，隐含假定了预测分布是高斯型的，即 $p(\boldsymbol{x}_k^{(i)} \mid \boldsymbol{y}_{1:k-1})$ 可近似为

$$
p(\boldsymbol{x}_k^{(i)} \mid \boldsymbol{y}_{1:k-1}) = \mathcal{N}(\boldsymbol{x}_k^{(i)}; \hat{\boldsymbol{x}}_{k|k-1}^{(i)}, \boldsymbol{P}_{k|k-1}^{(i)}) \tag{8-43}
$$

其中，状态预测及相应的协方差分别为

$$
\begin{aligned}
\hat{\boldsymbol{x}}_{k|k-1}^{(i)} &= E[\boldsymbol{x}_k^{(i)} \mid \boldsymbol{y}_{1:k-1}] = \int \boldsymbol{x}_k^{(i)} p(\boldsymbol{x}_k^{(i)} \mid \boldsymbol{y}_{1:k-1}) \mathrm{d}\boldsymbol{x}_k^{(i)} \\
&= \int \left[\int \boldsymbol{x}_k^{(i)} \mathcal{N}(\boldsymbol{x}_k^{(i)}; f(\boldsymbol{x}_{k-1}^{(i)}), \boldsymbol{Q}_k) \mathrm{d}\boldsymbol{x}_k^{(i)} \right] p(\boldsymbol{x}_{k-1}^{(i)} \mid \boldsymbol{y}_{1:k-1}) \mathrm{d}\boldsymbol{x}_{k-1}^{(i)} \\
&= \int f(\boldsymbol{x}_{k-1}^{(i)}) p(\boldsymbol{x}_{k-1}^{(i)} \mid \boldsymbol{y}_{1:k-1}) \mathrm{d}\boldsymbol{x}_{k-1}^{(i)} = \int f(\boldsymbol{x}_{k-1}^{(i)}) \mathcal{N}(\boldsymbol{x}_{k-1}^{(i)}; \hat{\boldsymbol{x}}_{k-1}^{(i)}, \boldsymbol{P}_{k-1}^{(i)}) \mathrm{d}\boldsymbol{x}_{k-1}^{(i)}
\end{aligned} \tag{8-44}
$$

$$
\begin{aligned}
\boldsymbol{P}_{k|k-1}^{(i)} &= E[(\boldsymbol{x}_k^{(i)} - \hat{\boldsymbol{x}}_{k|k-1}^{(i)})(\boldsymbol{x}_k^{(i)} - \hat{\boldsymbol{x}}_{k|k-1}^{(i)})^{\mathrm{T}}] \\
&= \boldsymbol{Q}_k + \int [f(\boldsymbol{x}_{k-1}^{(i)}) - \hat{\boldsymbol{x}}_{k|k-1}^{(i)}][f(\boldsymbol{x}_{k-1}^{(i)}) - \hat{\boldsymbol{x}}_{k|k-1}^{(i)}]^{\mathrm{T}} \mathcal{N}(\boldsymbol{x}_{k-1}^{(i)}; \hat{\boldsymbol{x}}_{k-1}^{(i)}, \boldsymbol{P}_{k-1}^{(i)}) \mathrm{d}\boldsymbol{x}_{k-1}^{(i)}
\end{aligned} \tag{8-45}
$$

同样，有

$$
\begin{aligned}
\hat{\boldsymbol{y}}_{k|k-1}^{(i)} &= E[\boldsymbol{y}_k^{(i)} \mid \boldsymbol{x}_k^{(i)}, \boldsymbol{y}_{1:k-1}] = \int \boldsymbol{y}_k^{(i)} p(\boldsymbol{y}_k^{(i)} \mid \boldsymbol{x}_k^{(i)}, \boldsymbol{y}_{1:k-1}) \mathrm{d}\boldsymbol{x}_k^{(i)} \\
&= \int h(\boldsymbol{x}_k^{(i)}) p(\boldsymbol{y}_k^{(i)} \mid \boldsymbol{x}_k^{(i)}, \boldsymbol{y}_{1:k-1}) \mathrm{d}\boldsymbol{x}_k^{(i)} = \int h(\boldsymbol{x}_k^{(i)}) \mathcal{N}(\boldsymbol{x}_k^{(i)}; \hat{\boldsymbol{x}}_{k|k-1}^{(i)}, \boldsymbol{P}_{k|k-1}^{(i)}) \mathrm{d}\boldsymbol{x}_k^{(i)}
\end{aligned} \tag{8-46}
$$

$$
\begin{aligned}
\boldsymbol{P}_{y_k y_k}^{(i)} &= E[(\boldsymbol{y}_k^{(i)} - \hat{\boldsymbol{y}}_{k|k-1}^{(i)})(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_{k|k-1}^{(i)})^{\mathrm{T}}] \\
&= \boldsymbol{R}_k + \int [h(\boldsymbol{x}_k^{(i)}) - \hat{\boldsymbol{y}}_{k|k-1}^{(i)}][h(\boldsymbol{x}_k^{(i)}) - \hat{\boldsymbol{y}}_{k|k-1}^{(i)}]^{\mathrm{T}} \mathcal{N}(\boldsymbol{x}_k^{(i)}; \hat{\boldsymbol{x}}_{k|k-1}^{(i)}, \boldsymbol{P}_{k|k-1}^{(i)}) \mathrm{d}\boldsymbol{x}_k^{(i)}
\end{aligned} \tag{8-47}
$$

$$
\begin{aligned}
\boldsymbol{P}_{x_k y_k}^{(i)} &= E[(\boldsymbol{x}_k^{(i)} - \hat{\boldsymbol{x}}_{k|k-1}^{(i)})(\boldsymbol{y}_k^{(i)} - \hat{\boldsymbol{y}}_{k|k-1}^{(i)})^{\mathrm{T}}] \\
&= \int (\boldsymbol{x}_k^{(i)} - \hat{\boldsymbol{x}}_{k|k-1}^{(i)})(h(\boldsymbol{x}_k^{(i)}) - \hat{\boldsymbol{y}}_{k|k-1}^{(i)})^{\mathrm{T}} \mathcal{N}(\boldsymbol{x}_k^{(i)}; \hat{\boldsymbol{x}}_{k|k-1}^{(i)}, \boldsymbol{P}_{k|k-1}^{(i)}) \mathrm{d}\boldsymbol{x}_k^{(i)}
\end{aligned} \tag{8-48}
$$

因此

$$\boldsymbol{K}_k^{(i)} = \boldsymbol{P}_{x_k y_k}^{(i)} (\boldsymbol{P}_{y_k y_k}^{(i)})^{-1} \tag{8-49}$$

$$\hat{\boldsymbol{x}}_k^{(i)} = \hat{\boldsymbol{x}}_{k|k-1}^{(i)} + \boldsymbol{K}_k^{(i)} (\boldsymbol{y}_k - \hat{\boldsymbol{y}}_{k|k-1}^{(i)}) \tag{8-50}$$

$$\boldsymbol{P}_k^{(i)} = \boldsymbol{P}_{k|k-1}^{(i)} - \boldsymbol{K}_k^{(i)} \boldsymbol{P}_{y_k y_k}^{(i)} (\boldsymbol{K}_k^{(i)})^{\mathrm{T}} \tag{8-51}$$

式（8-44）～式（8-48）所示的解析表达式是为了说明问题的求解方法。在实际应用中，这些式子的计算仍采用与式（8-23）～式（8-29）相似的 UT 方法来实现。如此一来，重要权的计算式可表示为

$$w_k^{(i)} = \frac{p(\boldsymbol{x}_k^{(i)} \mid \boldsymbol{y}_{1:k})}{\pi_{\mathcal{N}}(\boldsymbol{x}_k^{(i)} \mid \boldsymbol{y}_{1:k})} \propto \frac{p(\boldsymbol{y}_k \mid \boldsymbol{x}_k^{(i)}) p(\boldsymbol{x}_k^{(i)} \mid \boldsymbol{y}_{1:k-1})}{\pi_{\mathcal{N}}(\boldsymbol{x}_k^{(i)} \mid \boldsymbol{y}_{1:k})} \tag{8-52}$$

由此可知，当从建议分布 $\pi_{\mathcal{N}}(\boldsymbol{x}_k^{(i)} \mid \boldsymbol{y}_{1:k}) = \mathcal{N}(\boldsymbol{x}_k; \hat{\boldsymbol{x}}_k^{(i)}, \boldsymbol{P}_k^{(i)})$ 进行采样得到粒子 $\boldsymbol{x}_k^{(i)}$ 之后，相应的权值为

$$w_k^{(i)} \propto \frac{p(\boldsymbol{y}_k \mid \boldsymbol{x}_k^{(i)}) p(\boldsymbol{x}_k^{(i)} \mid \boldsymbol{y}_{1:k-1})}{\mathcal{N}(\boldsymbol{x}_k^{(i)}; \hat{\boldsymbol{x}}_k^{(i)}, \boldsymbol{P}_k^{(i)})} \approx \frac{p(\boldsymbol{y}_k \mid \boldsymbol{x}_k^{(i)}) \mathcal{N}(\boldsymbol{x}_k^{(i)}; \hat{\boldsymbol{x}}_{k|k-1}^{(i)}, \boldsymbol{P}_{k|k-1}^{(i)})}{\mathcal{N}(\boldsymbol{x}_k^{(i)}; \hat{\boldsymbol{x}}_k^{(i)}, \boldsymbol{P}_k^{(i)})} \tag{8-53}$$

在得到 k 时刻所有粒子的权值之后，该时刻总的后验概率密度可计算为

$$p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k}) \approx \sum_{i=1}^{N} \tilde{w}_k^{(i)} \mathcal{N}(\boldsymbol{x}_k; \hat{\boldsymbol{x}}_k^{(i)}, \boldsymbol{P}_k^{(i)}) \tag{8-54}$$

式中：$\tilde{w}_k^{(i)}$ 为归一化权重。

几点说明：

（1）这里所提的修正的 UPF（MUPF）算法可视为重要采样方法和高斯混合近似的结合，该方法与文献[37]～文献[39]中的高斯和滤波器（GSF）有明显的不同。在 GSF 中，每一个高斯分量通过 EKF 来更新，且相应的权值单纯由似然计算得到，即 $w_k^{(i)} = \mathcal{N}(\boldsymbol{y}_k - h(\hat{\boldsymbol{x}}_{k|k-1}^{(i)}); \boldsymbol{0}, \boldsymbol{P}_{k|k-1}^{(i)})$。

（2）重要采样方法的应用实例在文献[40]提出的高斯和粒子滤波器（GSPF）中已有介绍，但这里所提的 MUPF 算法与 GSPF 仍有着本质的差异。在 GSPF 中，每一个高斯分量的预测均值和协方差是通过蒙特卡罗仿真的方法得到的，这意味着需要产生大量的粒子来计算状态均值和方差。与此不同的是，对于 UPF 和 MUPF 而言，前二阶矩的更新是通过 UKF 算法来实现的。粒子从混合高斯分布进行采样，且每个高斯分量仅产生一个粒子。

（3）在式（8-34）中，权值通过先验计算递推求得；与此相比，式（8-53）中的修正权值则通过预测分布进行估算。由于预测协方差和滤波协方差是可比拟的（即在量值上相近），故数值敏感问题将得以消除，粒子权值的分布特性也同时得以改善。这反过来可以有效避免粒子退化问题，从而大大提高滤波器的估计性能。

8.4 应用实例

8.4.1 IPF的只测角跟踪

考虑二维平面内的只测角跟踪（BOT）问题[1]，目标运动的状态方程满足

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}\boldsymbol{x}_k + \boldsymbol{w}_k \tag{8-55}$$

式中：状态 $\boldsymbol{x}_k = [x_k\ \dot{x}_k\ y_k\ \dot{y}_k]^{\mathrm{T}}$；矩阵 $\boldsymbol{F} = \mathrm{diag}\{\boldsymbol{F}_1, \boldsymbol{F}_1\}$；$T$ 为观测时间间隔；$\boldsymbol{w}_k$ 为扰动噪声，其协方差阵 $\boldsymbol{Q}_k = \mathrm{diag}\{\boldsymbol{Q}_1, \boldsymbol{Q}_1\}$。另外，$\boldsymbol{F}_1 = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$，$\boldsymbol{Q}_1 = q^2 \begin{bmatrix} T^4/3 & T^3/2 \\ T^3/2 & T^2 \end{bmatrix}$。

观测量为目标方位角，其观测方程为

$$\varphi_k = \arctan(x_k / y_k) + v_k, \quad k = 1, 2, \cdots, K \tag{8-56}$$

式中：v_k 为零均值高斯噪声，即 $v_k \sim \mathcal{N}(0, \sigma_v^2)$。

仿真中取 T=1s，$q = 0.001^2$，$\sigma_v = 0.005$。目标初始状态为 $\boldsymbol{x}_0 = [-0.05, 0.001, 0.7, -0.055]^{\mathrm{T}}$，$K = 40$；状态先验分布 $p(\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_0; \hat{\boldsymbol{x}}_0, \boldsymbol{P}_0)$，其中 $\hat{\boldsymbol{x}}_0 = [-0.06, 0.0015, 0.65, -0.05]^{\mathrm{T}}$，$\boldsymbol{P}_0 = \mathrm{diag}\{0.02^2, 0.001^2, 0.05^2, 0.01^2\}$。目标跟踪的效果用航迹的均方误差（MSE）来度量，其定义为

$$MSE = \frac{1}{K}\sum_{k=1}^{K}[(x_k - \hat{x}_k)^2 + (y_k - \hat{y}_k)^2] \tag{8-57}$$

图 8.2 为几种不同算法（EKF、UKF、SPF 和 IPF）对航迹进行跟踪的比较示例，图 8.3 为 100 次独立实验的目标航迹 MSE 比较，两图中 N=1000。

由图 8.2 中可以看出，基于粒子滤波的跟踪性能比 EKF 和 UKF 有了较大的改善，这不仅体现在对状态的估计精度上，同时也体现在对目标的跟踪能力上。由图 8.3 可以清楚地看出，在成功跟踪的前提下，EKF 和 UKF 的跟踪误差基本上在同一量级，但这并不意味着两者性能相近。应当看到，在 100 次实验中 EKF 有 26 次 MSE 超过 1，这说明，这些跟踪产生了严重的发散。相对而言，UKF 要好得多，图 8.3 中所示 MSE 超过 1 的仅有 1 次。

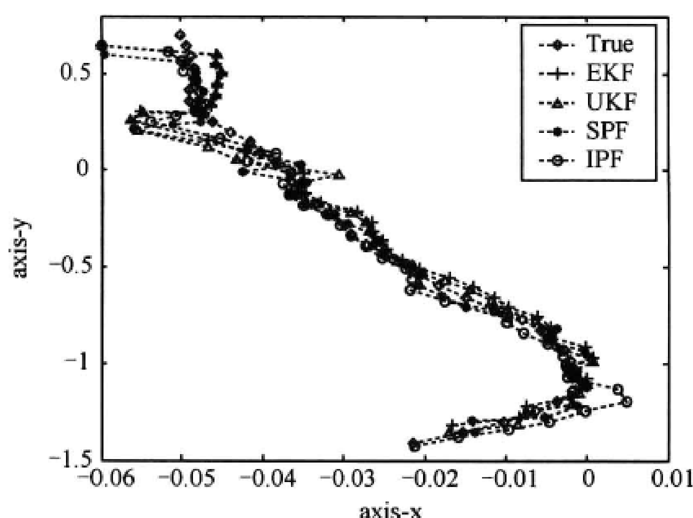


图 8.2 目标航迹及其估计

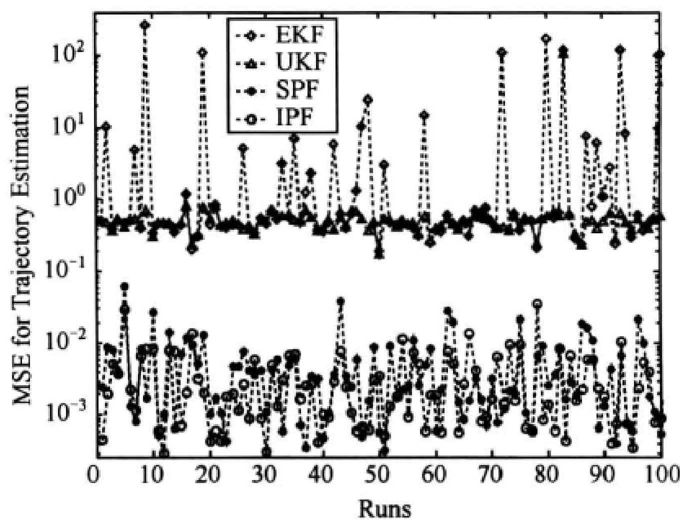


图 8.3 100 次独立试验 MSE 比较

很明显，基于粒子滤波的跟踪算法都没出现明显的跟踪发散现象，其估计误差也要小得多。为了更好地比较 SPF 和 IPF 两种粒子滤波器的性能，将不同条件下的仿真结果列于表 8.1 中。需要说明的是，由于 EKF 和 UKF 两种方法都有发散现象，对 MSE 作简单的平均比较并无实际意义，因此在表中并未列出。

表 8.1 不同粒子数目条件下的仿真结果比较

Algorithms		MSE	
		Mean（均值）	Std（标准差）
SPF	N=500	0.005013	0.004409
	N=1000	0.003779	0.003348
	N=1500	0.003582	0.002833
	N=2000	0.003203	0.002677
IPF	N=500	0.002460	0.001948
	N=1000	0.002161	0.001864
	N=1500	0.002027	0.001713
	N=2000	0.001996	0.001605

图 8.4 对两种算法在 Matlab 仿真环境下的平均运行时间作了比较。图中给出的是程序实际运行时间对 SPF 算法在 N=500 条件下运行时间的归一化结果。从图中的结果看，在粒子数相同的前提下，IPF 的运行时间比 SPF 有所增加，前者约为后者的 2.2 倍（若对 IPF 算法进行某种优化，可进一步降低相对运行时间）。对比表 8.1 的结果可以发现，在运算时间开销相同的条件下，IPF 的估计结果仍要优于 SPF。

总之，IPF 算法通过对最优采样函数的合理近似，能产生更有效的粒子来逼近真实的状态分布；同时，通过 MCMC 策略来减轻重采样可能导致的粒子贫化现象，从而提高算法的估计性能。对只测角跟踪的仿真结果表明，IPF 算法比 EKF 和 UKF 具有更高的精度和更强的鲁棒性；与 SPF 相比，其性能也有较大的提高，并可以在相同的估计精度要求下大大减少所需的粒子数目。

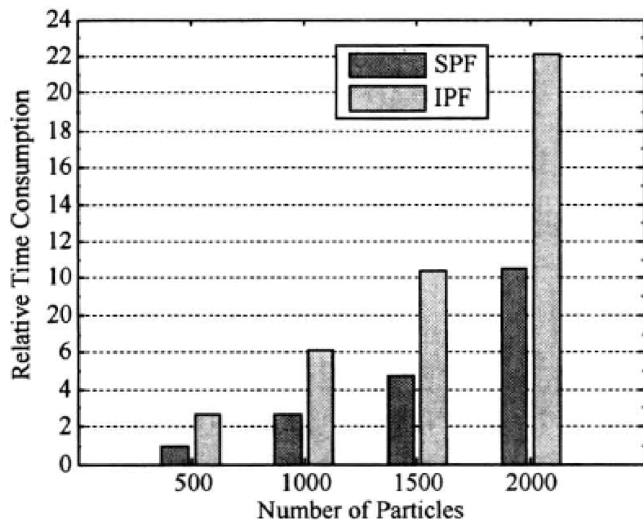


图 8.4 两种算法时间复杂度比较

8.4.2 MUPF 的被动跟踪

1. 只测角跟踪

仍以 8.4.1 节中的只测角跟踪为例，验证跟踪算法的性能。为区别起见，这里将采用更一般的过程噪声形式。目标的状态方程为

$$x_k = Fx_{k-1} + \Gamma w_k \tag{8-58}$$

式中

$$x_k = [x_k\ y_k\ \dot{x}_k\ \dot{y}_k]^{\mathrm{T}}，\quad w_k = [w_x\ w_y]^{\mathrm{T}}，\quad F = \begin{bmatrix} I_2 & T \cdot I_2 \\ 0 & I_2 \end{bmatrix}，\quad \Gamma = \begin{bmatrix} 0.5T^2 \cdot I_2 \\ T \cdot I_2 \end{bmatrix}$$

过程噪声为 $w_k \sim \mathcal{N}(0,Q)$，$Q = 0.001^2 \cdot I_2$；观测方程及其他仿真参数设置与前一实例保持一致。在不同的粒子数目条件下，分别用 UPF 和 MUPF 对目标的状态进行估计。算法性能用 MSE 和平均有效采样数（Averaged Effective Sample Number，AESN）来评估，AESN 定义为

$$AESN = \frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{N}(\tilde{w}_k^{(i)})^{-2} \tag{8-59}$$

式中：K 为每次仿真的观测次数；N 为粒子数。

注意到协方差矩阵 $\Gamma Q \Gamma^{\mathrm{T}}$ 的奇异性，先验 $p(x_k \mid x_{k-1})$ 不能通过式（8-34）直接计算得到，故在传统 UPF 算法中必须采取特殊的措施来处理这一问题。一种可能的方法是将高维状态空间问题转化为低维噪声空间问题来处理。由式（8-58）可得

$$p(x_k \mid x_{k-1}) = p_w(G(x_k - Fx_{k-1})) \tag{8-60}$$

式中：$G = (\Gamma^{\mathrm{T}}\Gamma)^{-1}\Gamma^{\mathrm{T}}$。经过这种转化，就可以用相对简单的方法计算粒子的等效权重。

图 8.5 为两种滤波器对目标进行单次跟踪的示例；图 8.6 为 100 次独立实验中目标位置估计的 MSE，粒子个数为 200；图 8.7 为 100 次仿真中 AESN 的变化情况，图 8.8 比较了不同粒子数目条件下，两种算法 MSE 的均值和标准偏差。

如图 8.5～图 8.8 所示，在相同的仿真条件下，MUPF 要明显优于 UPF。究其原因，主要是因为 MUPF 算法能有效改善粒子的权值分布特性，使粒子的多样性得以有效保持

（这从图 8.7 可以清楚地看出），从而有效提高滤波器的性能。由图 8.8 可知，MUPF 算法的 MSE 均值和标准偏差都较 UPF 要小，这说明 MUPF 不仅有更优的估计精度，算法的稳定性也得以提高，收敛一致性也更好。

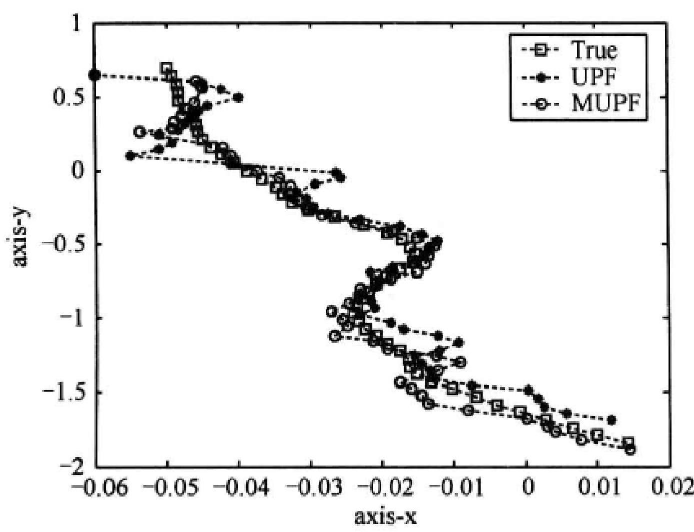


图 8.5　目标真实位置与位置估计（N=200）

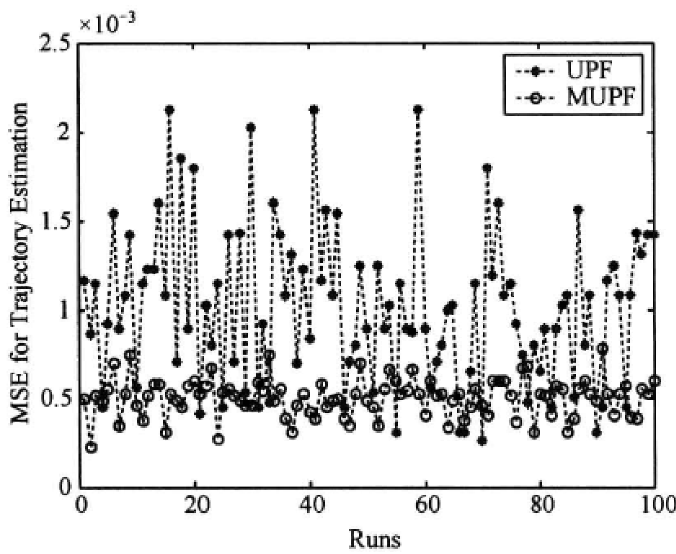


图 8.6　100 次独立实验位置 MSE（N=200）

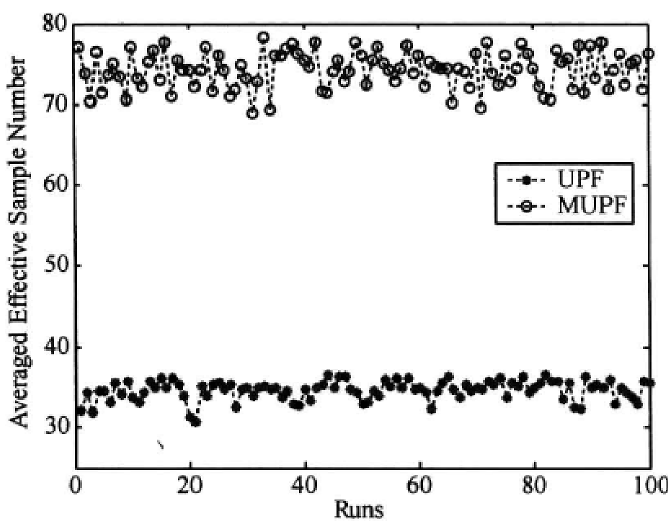


图 8.7　100 次仿真中平均有效粒子数

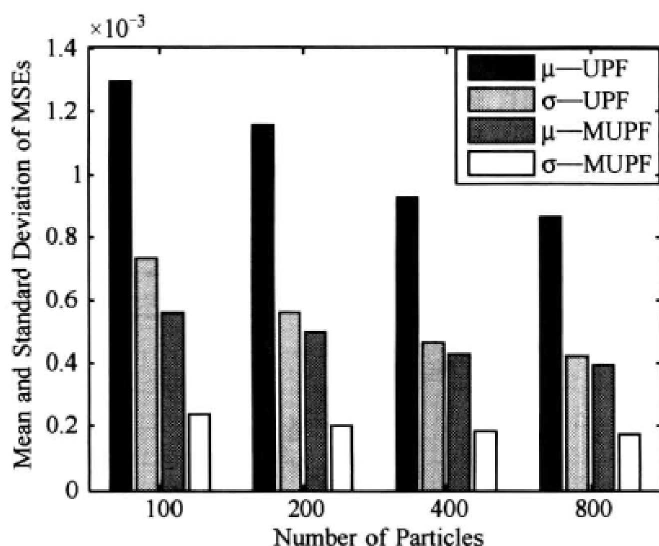


图 8.8　目标位置 MSE 均值和标准偏差

2. 方位—频率联合跟踪

假定目标初始位置和速度分别为 (50, 80) km 和 (−150,50)m/s，过程噪声协方差矩阵 $\boldsymbol{Q}_k = \text{diag}\{(2\text{m/s}^2)^2,(2\text{m/s}^2)^2\}$。进一步假定 T=1s，f_0=3GHz（多普勒频率观测模型见 9.3.1 节），观测精度为{3mrad，10Hz}。采用 UPF 和 MUPF 分别对其跟踪，粒子数目 N=400，仿真次数为 100，所得结果如图 8.9 和图 8.10 所示。

由图 8.9 和图 8.10 可以看出，两种滤波器都能以较快的速度收敛。但相比之下，MUPF 无论在收敛速度还是跟踪精度方面都要优于 UPF，表明前者能得到更准确的状态估计。本例中的仿真结果与只测角跟踪的结论是完全一致的，这就说明，MUPF 在稳定性、跟踪速度和精度等方面确实比 UPF 有了明显的改善，是一种更为有效的跟踪算法。

综合两个仿真实例的结果可知，MUPF 算法能有效解决传统 UPF 中的潜在不足，通过对权值的修正可以明显改善其分布特性，从而使粒子的多样性得以保持。概言之，MUPF 与传统 UPF 算法相比具有两大潜在优势：一是在同样的估计性能要求下，可以大大减小粒子数目，从而有效降低运算开销；另一个优点在于当过程噪声协方差矩阵奇异时，对重要权的计算无需采取任何特殊措施，使得 MUPF 算法更便于实现。此外，从整个算法流程可以看出，两种算法具有相同的复杂度，这一复杂度约为 UKF 的 N 倍（N 为粒子数）。

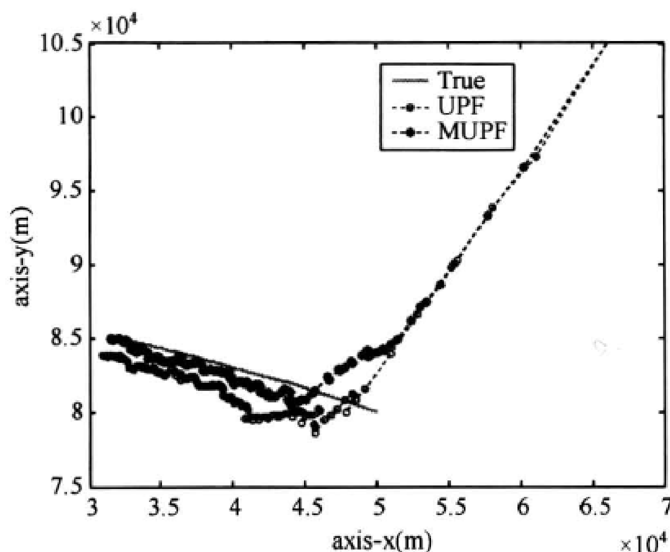


图 8.9　两种算法跟踪结果比较

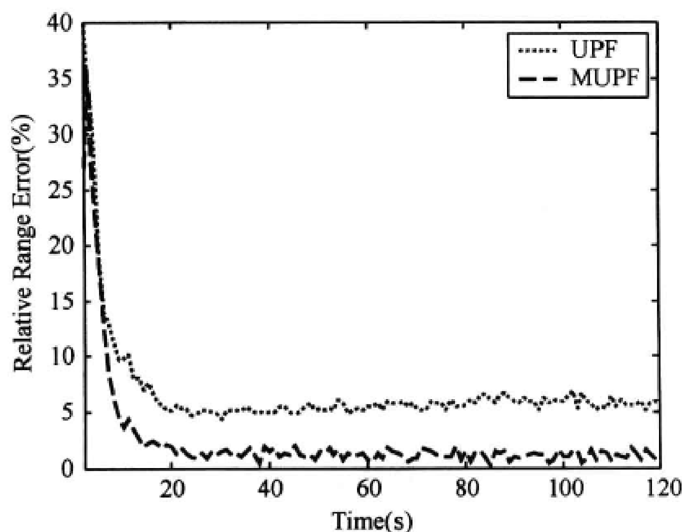


图 8.10 相对距离误差比较

8.5 本章小结

本章主要阐述基于序贯重要采样的粒子滤波原理及其在目标跟踪中的应用问题。粒子滤波的关键是如何在给定算法复杂度的条件下得到真实后验状态分布的有效近似，传统粒子滤波器直接根据状态转移方程产生粒子，由于没有考虑量测的修正影响，估计性能较差。针对这一问题，提出了一种通过最优采样函数近似生成粒子的方法，并加入MCMC 过程来保持粒子的多样性，从而提高了滤波器的估计性能。

UPF 算法通过 UKF 的滤波结果产生建议分布，由于 UKF 具有良好的性能，UPF 被认为是一种较为有效的滤波算法。但在被动跟踪背景下，跟踪误差与噪声方差相差悬殊，容易出现数值敏感问题，导致算法性能严重下降。为此文中提出一种修正算法——MUPF，通过对权值的修正使其避免数值问题，并有效减轻了粒子贫化现象，从而提高滤波效果。

值得一提的是，理论上，本书第二部分阐述的其他形式的高斯滤波算法都可用做指导粒子生成的建议分布，由此得到各种基于重要性采样函数的粒子滤波实现形式，由于其原理相似，在此不再赘述。另外，除了前文所述的几种建议分布函数形式外，目前还存在一些其他形式，感兴趣的读者可参阅文献[41]。

第 9 章　边缘化采样滤波

尽管从前面的仿真结果可以发现，粒子滤波算法比传统滤波算法如 EKF、UKF 等有着更广泛的适用性和更强的跟踪能力，但同时也清楚地看到，这些优势的获得是以增加运算量为代价的。理论上，在同等近似精度要求下，粒子滤波算法中所需的粒子数与状态量的维数呈指数增长，这一运算量对许多高维状态空间的估计问题是难以承受的。

但在很多特殊的情况下（包括目标跟踪应用场合），系统包含一个线性的子结构，即系统模型中部分状态可用线性模型来描述。这就意味着，系统的线性部分可以用最优卡尔曼滤波进行估计，而粒子只需存在于低维的非线性状态空间，从而在给定粒子数目的条件下使估计性能得以改善。这种从总的系统状态 PDF 中提取线性状态分布的方法称为“边缘化”，相应的估计算法称为边缘化粒子滤波（Marginalized Particle Filtering）[42-43]或 Rao-Blackwellized 粒子滤波[44-46]。

本章首先介绍了混合系统状态（这里特指线性与非线性部分的混合）通用的边缘化粒子滤波处理思路，并给出了相应的算法流程；在此基础上重点针对几种重要的系统模型（如对角模型、三角模型及一类与目标跟踪密切相关的特殊模型），详细阐述了其边缘化滤波方法，并分析了算法的复杂度，最后给出了算法的目标跟踪应用实例。

9.1　混合状态的边缘化

如前所述，粒子滤波方法可以递推地产生随机采样点来近似后验 PDF $p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k})$，为非线性、非高斯系统中的状态估计问题建立了统一的处理框架。但由于众所周知的原因，当状态量的维数比较高时，必须产生大量的粒子才能较好地近似高维状态的 PDF，导致 PF 在许多实际情况中并不适用。

然而，在很多特定情况下，系统模型都具有可解析求解的子结构，因此在非线性状态量已知的条件下，可以将线性状态“边缘化”出来，使 PF 只工作于较低维的非线性状态空间。其基本思路是将状态变量分成两部分，即

$$\boldsymbol{x}_k = [(\boldsymbol{x}_k^n)^{\mathrm{T}}\ (\boldsymbol{x}_k^l)^{\mathrm{T}}]^{\mathrm{T}} \tag{9-1}$$

式中：$\boldsymbol{x}_k^l$ 为依赖于系统条件的线性状态量；$\boldsymbol{x}_k^n$ 为非线性状态量。若令 $\boldsymbol{x}_{0:k}^n = \{\boldsymbol{x}_i^n\}_{i=0}^{i=k}$，则由贝叶斯定理易得

$$p(\boldsymbol{x}_k^l, \boldsymbol{x}_{0:k}^n \mid \boldsymbol{y}_{1:k}) = p(\boldsymbol{x}_k^l \mid \boldsymbol{x}_{0:k}^n, \boldsymbol{y}_{1:k})p(\boldsymbol{x}_{0:k}^n \mid \boldsymbol{y}_{1:k}) \tag{9-2}$$

式中：$p(\boldsymbol{x}_k^l \mid \boldsymbol{x}_{0:k}^n, \boldsymbol{y}_{1:k})$ 易于解析求解，可通过最优卡尔曼滤波器（KF）来实现；而 $p(\boldsymbol{x}_{0:k}^n \mid \boldsymbol{y}_{1:k})$ 则无法求得解析解，可通过 PF 来实现。

下面将给出边缘化滤波的一般处理流程，总结如下。

Alg（9.1）：Generic Marginalized Particle Filtering

步骤 1：初始化

$$\text{For}\quad i=1,2,\cdots,N$$

对粒子 $x_{0|-1}^{n,(i)} \sim p(x_0^n)$ 和集合 $\{x_{0|-1}^{l,(i)}, P_{0|-1}^{(i)}\}=\{\hat{x}_{0|-1}^l, P_0\}$ 进行初始化。

步骤 2：粒子量测更新

$$\text{For}\quad i=1,2,\cdots,N$$

计算粒子权重 $w_k^{(i)}=p(y_k \mid x_{0:k}^{n,(i)}, y_{1:k-1})$ 并进行归一化：

$$\tilde{w}_k^{(i)}=w_k^{(i)}/\sum_{j=1}^{N} w_k^{(j)}$$

步骤 3：粒子重采样

对 N 个粒子进行重采样使得 $\Pr\left(x_k^{n,(i)}=x_{k|k-1}^{n,(j)}\right)=\tilde{w}_k^{(j)}$。

步骤 4：粒子滤波时间更新及卡尔曼滤波

（1）卡尔曼量测更新。

（2）粒子滤波时间更新。

$$\text{For}\quad i=1,2,\cdots,N$$

预测新粒子 $x_{k+1|k}^{n,(i)} \sim p(x_{k+1|k}^n \mid x_{0:k}^{n,(i)}, y_{1:k})$。

（3）卡尔曼时间更新。

步骤 5：$k=k+1$，从步骤 2 开始进行下一轮递推

9.2 边缘化粒子滤波算法

9.2.1 对角模型算法

考虑下式所示的系统模型

$$\begin{cases} x_k^n = f(x_{k-1}^n) + w_k^n \\ x_k^l = A_{k-1}^l(x_{k-1}^n)x_{k-1}^l + w_k^l \\ y_k = h(x_k^n) + C_k(x_k^n)x_k^l + v_k \end{cases} \tag{9-3}$$

式中：w_k 为过程噪声，且

$$w_k=\begin{bmatrix} w_k^l \\ w_k^n \end{bmatrix} \sim \mathcal{N}(0, Q_k),\quad Q_k=\begin{bmatrix} Q_k^l & 0 \\ 0 & Q_k^n \end{bmatrix} \tag{9-4}$$

v_k 为量测噪声且 $v_k \sim \mathcal{N}(0, R_k)$；$x_0^l$ 服从高斯分布且 $x_0^l \sim \mathcal{N}(\hat{x}_0, P_0)$；$x_0^n$ 的分布可任意但必须已知，A^l、C 为任意矩阵。

由于式（9-3）表示的模型具有类似于对角结构的特点，因此该模型可称为对角模型。贝叶斯滤波估计的目的是利用粒子滤波算法得到后验密度 $p(x_k \mid y_{1:k})$，但是由式（9-3）可知，在 x_k^n 已知的条件下，该式存在一个线性的子结构。根据式（9-2）表示的边缘化公式，$p(x_k^l \mid x_{0:k}^n, y_{1:k})$ 可通过卡尔曼滤波解析求解，而 $p(x_{0:k}^n \mid y_{1:k})$ 则可通过粒子滤波求解。可以预见，在粒子数相同的条件下，采用边缘化粒子滤波得到的估计比传统的粒子

滤波估计效果要好，这主要归结为两方面的原因：一是 $p(\boldsymbol{x}_k^n \mid \boldsymbol{y}_{1:k})$ 的维数比 $p(\boldsymbol{x}_k^l,\boldsymbol{x}_k^n \mid \boldsymbol{y}_{1:k})$ 要低，表明粒子处于较低维的空间，因而近似精度更高；二是在估计线性状态量时采用了最优算法，相应地性能也将会更佳。对于任意非线性函数 $g(\cdot)$，若定义

$$I\left(g(\boldsymbol{x}_k)\right)=E_{p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})}\left(g(\boldsymbol{x}_k)\right)=\int g(\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})\mathrm{d}\boldsymbol{x}_k \tag{9-5}$$

并记 $\hat{I}_N^s\left(g(\boldsymbol{x}_k)\right)$ 和 $\hat{I}_N^m\left(g(\boldsymbol{x}_k)\right)$ 分别为利用 N 个粒子得到的对式（9-5）的标准粒子滤波和边缘化粒子滤波估计，则当 $N\to\infty$ 时，根据中心极限定理有下列结论成立，即

$$\sqrt{N}[\hat{I}_N^s\left(g(\boldsymbol{x}_k)\right)-I\left(g(\boldsymbol{x}_k)\right)]\Rightarrow\mathcal{N}(0,\sigma_s^2) \tag{9-6}$$

$$\sqrt{N}[\hat{I}_N^m\left(g(\boldsymbol{x}_k)\right)-I\left(g(\boldsymbol{x}_k)\right)]\Rightarrow\mathcal{N}(0,\sigma_m^2) \tag{9-7}$$

式中：$\sigma_s^2\geqslant\sigma_m^2$。关于该式的详细证明见文献[7]。

下面详细分析 $p(\boldsymbol{x}_k^l \mid \boldsymbol{x}_{0:k}^n,\boldsymbol{y}_{1:k})$ 与 $p(\boldsymbol{x}_{0:k}^n \mid \boldsymbol{y}_{1:k})$ 的求解过程。事实上，对于第一项，由卡尔曼滤波公式，易得 $\boldsymbol{x}_k^l$ 和 $\boldsymbol{x}_{k+1|k}^l$ 的条件概率密度函数为

$$p(\boldsymbol{x}_k^l \mid \boldsymbol{x}_{0:k}^n,\boldsymbol{y}_{1:k})=\mathcal{N}(\boldsymbol{x}_k^l;\hat{\boldsymbol{x}}_k^l,\boldsymbol{P}_k) \tag{9-8}$$

$$p(\boldsymbol{x}_{k+1}^l \mid \boldsymbol{x}_{0:k+1}^n,\boldsymbol{y}_{1:k})=\mathcal{N}(\boldsymbol{x}_{k+1}^l;\hat{\boldsymbol{x}}_{k+1|k}^l,\boldsymbol{P}_{k+1|k}) \tag{9-9}$$

式中

$$\hat{\boldsymbol{x}}_k^l=\hat{\boldsymbol{x}}_{k|k-1}^l+\boldsymbol{K}_k(\boldsymbol{y}_k-h_k-\boldsymbol{C}_k\hat{\boldsymbol{x}}_{k|k-1}^l) \tag{9-10}$$

$$\boldsymbol{P}_k=\boldsymbol{P}_{k|k-1}-\boldsymbol{K}_k\boldsymbol{C}_k\boldsymbol{P}_{k|k-1} \tag{9-11}$$

$$\boldsymbol{K}_k=\boldsymbol{P}_{k|k-1}\boldsymbol{C}_k^{\mathrm{T}}\boldsymbol{S}_k^{-1} \tag{9-12}$$

$$\boldsymbol{S}_k=\boldsymbol{C}_k\boldsymbol{P}_{k|k-1}\boldsymbol{C}_k^{\mathrm{T}}+\boldsymbol{R}_k \tag{9-13}$$

$$\hat{\boldsymbol{x}}_{k|k-1}^l=\boldsymbol{A}_{k-1}^l\hat{\boldsymbol{x}}_{k-1}^l \tag{9-14}$$

$$\boldsymbol{P}_{k|k-1}=\boldsymbol{A}_{k-1}^l\boldsymbol{P}_{k-1}(\boldsymbol{A}_{k-1}^l)^{\mathrm{T}}+\boldsymbol{Q}_k^l \tag{9-15}$$

上述递推式的初始条件为 $\boldsymbol{x}_{0|-1}^l=\hat{\boldsymbol{x}}_0$，$\boldsymbol{P}_{0|-1}=\boldsymbol{P}_0$。

对于第二项 $p(\boldsymbol{x}_{0:k}^n \mid \boldsymbol{y}_{1:k})$，根据贝叶斯理论，有

$$\begin{aligned}p(\boldsymbol{x}_{0:k}^n \mid \boldsymbol{y}_{1:k})&=\frac{p(\boldsymbol{x}_{0:k}^n,\boldsymbol{y}_{1:k})}{p(\boldsymbol{y}_{1:k})}=\frac{p(\boldsymbol{x}_k^n,\boldsymbol{y}_k,\boldsymbol{x}_{0:k-1}^n,\boldsymbol{y}_{1:k-1})}{p(\boldsymbol{y}_k \mid \boldsymbol{y}_{1:k-1})p(\boldsymbol{y}_{1:k-1})}\\&=\frac{p(\boldsymbol{x}_k^n,\boldsymbol{y}_k \mid \boldsymbol{x}_{0:k-1}^n,\boldsymbol{y}_{1:k-1})}{p(\boldsymbol{y}_k \mid \boldsymbol{y}_{1:k-1})}p(\boldsymbol{x}_{0:k-1}^n \mid \boldsymbol{y}_{1:k-1})\\&=\frac{p(\boldsymbol{y}_k|\boldsymbol{x}_{0:k}^n,\boldsymbol{y}_{1:k-1})p(\boldsymbol{x}_k^n \mid \boldsymbol{x}_{0:k-1}^n,\boldsymbol{y}_{1:k-1})}{p(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1})}p(\boldsymbol{x}_{0:k-1}^n \mid \boldsymbol{y}_{1:k-1})\end{aligned} \tag{9-16}$$

式中：$p(\boldsymbol{x}_{0:k-1}^n \mid \boldsymbol{y}_{1:k-1})$ 可由前一时刻（$k-1$ 时刻）的粒子滤波估计得到；解析表达式 $p(\boldsymbol{y}_k|\boldsymbol{x}_{0:k}^n,\boldsymbol{y}_{1:k-1})$ 及 $p(\boldsymbol{x}_k^n \mid \boldsymbol{x}_{0:k-1}^n,\boldsymbol{y}_{1:k-1})$ 可表示为

$$p(\boldsymbol{y}_k|\boldsymbol{x}_{0:k}^n,\boldsymbol{y}_{1:k-1})=\mathcal{N}(\boldsymbol{y}_k;h_k+\boldsymbol{C}_k\boldsymbol{x}_{k+1}^l,\boldsymbol{C}_k\boldsymbol{P}_{k+1|k}\boldsymbol{C}_k^{\mathrm{T}}+\boldsymbol{R}_k) \tag{9-17}$$

$$p(\boldsymbol{x}_k^n \mid \boldsymbol{x}_{0:k-1}^n,\boldsymbol{y}_{1:k-1})=\mathcal{N}(\boldsymbol{x}_k^n;f_{k-1},\boldsymbol{Q}_{k-1}) \tag{9-18}$$

由此可见，式（9-3）的状态估计可通过粒子滤波 $\{\boldsymbol{x}_k^{n,(i)}\}_{i=1}^N$ 和卡尔曼滤波得到，对于每一个粒子，需要一个相应的卡尔曼滤波器。完整的滤波算法流程可见 Alg（9.1），由该

算法中可以看出，与标准的粒子滤波相比，MPF 算法在时间更新（预测）阶段发生了变化。在 SPF 算法中，预测过程仅由 Alg（9.1）中的步骤 4（2）来完成，步骤 4（1）在卡尔曼滤波中称为量测更新。另外，非线性状态量的预测 $\boldsymbol{x}_{k+1|k}^{n}$ 是通过步骤 4（2）得到的。根据式（9-3）中第一式可知，非线性状态量的预测不包含线性状态的任何信息，表明 $\boldsymbol{x}_{k+1|k}^{n}$ 不能用于提高线性状态量的估计质量。但是，若在式（9-3）第一式中线性状态与非线性状态之间具有某种依赖关系，则非线性状态可用于提高线性状态的估计效果，这一点将在 9.2.2 节中进行分析。

9.2.2 三角模型算法

将式（9-3）进行扩展并包含 $\boldsymbol{A}_k^n(\boldsymbol{x}_k^n)\boldsymbol{x}_k^l$ 项，即可得到如下三角模型：

$$\begin{cases}\boldsymbol{x}_k^n = f(\boldsymbol{x}_{k-1}^n)+\boldsymbol{A}_{k-1}^n(\boldsymbol{x}_{k-1}^n)\boldsymbol{x}_k^l+\boldsymbol{w}_k^n \\ \boldsymbol{x}_k^l = \boldsymbol{A}_{k-1}^l(\boldsymbol{x}_{k-1}^n)\boldsymbol{x}_{k-1}^l+\boldsymbol{w}_k^l \\ \boldsymbol{y}_k = h(\boldsymbol{x}_k^n)+\boldsymbol{C}_k(\boldsymbol{x}_k^n)\boldsymbol{x}_k^l+\boldsymbol{v}_k\end{cases} \tag{9-19}$$

由式（9-19）第一式可以看出，$\hat{\boldsymbol{x}}_{k|k-1}^n$ 包含了线性状态的信息，这意味着在非线性状态的预测 $\hat{\boldsymbol{x}}_{k|k-1}^n$ 中含有线性状态 $\boldsymbol{x}_{k-1}^l$。为了说明这一点，假定在 Alg（9.1）的步骤 4（2）中已经处理完成，即 $\hat{\boldsymbol{x}}_{k|k-1}^n$ 已经是可用的，此时该模型可写成

$$\begin{cases}\boldsymbol{x}_k^l=\boldsymbol{A}_{k-1}^l\boldsymbol{x}_{k-1}^l+\boldsymbol{w}_{k-1}^l \\ \boldsymbol{z}_k=\boldsymbol{A}_{k-1}^n\boldsymbol{x}_{k-1}^l+\boldsymbol{w}_{k-1}^n\end{cases} \tag{9-20}$$

式中：$\boldsymbol{z}_k=\boldsymbol{x}_{k+1}^l-\boldsymbol{f}_k^n$，可视为含噪声 $\boldsymbol{w}_{k-1}^n$ 的量测。

由于式（9-20）为线性状态空间模型，含有高斯分布噪声，其最优状态估计可由卡尔曼滤波得到且

$$\begin{aligned}&\hat{\boldsymbol{x}}_k^{l*}=\hat{\boldsymbol{x}}_k^l+\boldsymbol{L}_k(\boldsymbol{z}_k-\boldsymbol{A}_k^n\hat{\boldsymbol{x}}_k^l) \\ &\boldsymbol{P}_k^*=\boldsymbol{P}_k-\boldsymbol{L}_k\boldsymbol{N}_k\boldsymbol{L}_k^{\mathrm{T}} \\ &\boldsymbol{L}_k=\boldsymbol{P}_k(\boldsymbol{A}_k^{(n)})^{\mathrm{T}}\boldsymbol{N}_k^{-1} \\ &\boldsymbol{N}_k=\boldsymbol{A}_k^{(n)}\boldsymbol{P}_k(\boldsymbol{A}_k^{(n)})^{\mathrm{T}}+\boldsymbol{Q}_k^n\end{aligned} \tag{9-21}$$

式中：上标“*”用于与第一次量测更新进行区分。$\hat{\boldsymbol{x}}_k^l$ 和 $\boldsymbol{P}_k$ 可通过与式（9-10）和式（9-11）相类似的方法得到。

最后一步是将第二次量测更新与时间更新进行合并得到预测状态，其结果可表示为

$$\begin{aligned}&\hat{\boldsymbol{x}}_{k+1|k}^l=\boldsymbol{A}_k^l\hat{\boldsymbol{x}}_k^l+\boldsymbol{L}_k(\boldsymbol{z}_k-\boldsymbol{A}_k^n\hat{\boldsymbol{x}}_k^l) \\ &\boldsymbol{P}_{k+1|k}=\boldsymbol{A}_k^l\boldsymbol{P}_k(\boldsymbol{A}_k^l)^{\mathrm{T}}+\boldsymbol{Q}_k^l-\boldsymbol{L}_k\boldsymbol{N}_k\boldsymbol{L}_k^{\mathrm{T}} \\ &\boldsymbol{L}_k=\boldsymbol{A}_k^l\boldsymbol{P}_k(\boldsymbol{A}_k^{(n)})^{\mathrm{T}}\boldsymbol{N}_k^{-1} \\ &\boldsymbol{N}_k=\boldsymbol{A}_k^{(n)}\boldsymbol{P}_k(\boldsymbol{A}_k^{(n)})^{\mathrm{T}}+\boldsymbol{Q}_k^n\end{aligned} \tag{9-22}$$

式（9-22）的详细证明见文献[42]，为了使 Alg（9.1）中的算法适用于该模型，式（9-14）和式（9-15）中的时间更新方程必须用式（9-22）代替。

9.2.3 特殊模型算法

在实际应用中，存在一类重要的系统，具有线性状态方程和非线性量测方程，可用如下模型表示为

$$\begin{cases} \boldsymbol{x}_{k+1}^n = \boldsymbol{A}_k^n \boldsymbol{x}_k^n + \boldsymbol{A}_k^l \boldsymbol{x}_k^l + \boldsymbol{w}_k^n \\ \boldsymbol{x}_{k+1}^l = \boldsymbol{B}_k^n \boldsymbol{x}_k^n + \boldsymbol{B}_k^l \boldsymbol{x}_k^l + \boldsymbol{w}_k^l \end{cases} \tag{9-23}$$

$$\boldsymbol{y}_k = h(\boldsymbol{x}_k^n) + \boldsymbol{v}_k \tag{9-24}$$

式中：$\boldsymbol{w}_k^n \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_k^n)$；$\boldsymbol{w}_k^l \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_k^l)$；$\boldsymbol{v}_k$ 为已知分布的任意噪声。

上述模型广泛应用于描述直角坐标系中的目标运动及各种量测坐标系中的系统观测，在目标跟踪等领域具有重要的用途。在该模型中，式（9-24）中未包含任何关于线性状态 $\boldsymbol{x}_k^l$ 的信息，因此就卡尔曼滤波而言，式（9-24）不能用于估计线性状态；相反，所有的观测信息通过第二次量测更新隐含地融入到卡尔曼滤波中，这意味着 Alg（9.1）中的步骤 4（1）步骤可以省略。在这种条件下，第二次量测更新不仅仅只是第一次量测更新的修正，而是观测信息 $\boldsymbol{y}_k$ 融入算法中的唯一途径。

鉴于该模型的重要性，下面给出基于该模型的完整的滤波算法流程。

Alg（9.2）：Marginalized Particle Filtering （MPF）

步骤 1：对 N 个粒子进行初始化，使得 $\boldsymbol{x}_{0|-1}^{n,(i)} \sim p(\boldsymbol{x}_0^n)(i=1,2,\cdots,N)$ 且 $\{\boldsymbol{x}_{0|-1}^{l,(i)}, \boldsymbol{P}_{0|-1}^{(i)}\} = \{\bar{\boldsymbol{x}}_0^l, \boldsymbol{P}_0\}$。

步骤 2：计算权重 $w_k^{(i)} = p(\boldsymbol{y}_k \mid \boldsymbol{x}_{0:k}^n, \boldsymbol{y}_{1:k-1}) = p_v\left(\boldsymbol{y}_k - h(\boldsymbol{x}_k^n)\right)$ 并归一化得到 $\tilde{w}_k^{(i)}$。

步骤 3：对粒子重采样，使得 $\Pr(\boldsymbol{x}_k^{n,(i)} = \boldsymbol{x}_{k|k-1}^{n,(j)}) = \tilde{w}_k^{(j)}$。

步骤 4：粒子时间更新和卡尔曼滤波更新。

① 卡尔曼量测更新，即

$$\hat{\boldsymbol{x}}_k^{l,(i)} = \hat{\boldsymbol{x}}_{k|k-1}^{l,(i)}, \quad \boldsymbol{P}_k = \boldsymbol{P}_{k|k-1}$$

② 粒子时间更新（预测），即

$$\begin{aligned} \boldsymbol{x}_{k+1|k}^{n,(i)} &\sim p(\boldsymbol{x}_{k+1}^n \mid \boldsymbol{x}_{0:k}^{n,(i)}, \boldsymbol{y}_{1:k}) \\ &= \mathcal{N}\left(\boldsymbol{x}_{k+1}^n; \boldsymbol{A}_k^n \boldsymbol{x}_k^{n,(i)} + \boldsymbol{A}_k \hat{\boldsymbol{x}}_k^{l,(i)}, \boldsymbol{Q}_k^n + \boldsymbol{A}_k^l \boldsymbol{P}_k (\boldsymbol{A}_k^l)^{\mathrm{T}}\right) \end{aligned}$$

③ 卡尔曼滤波时间更新，即

$$\begin{aligned} \hat{\boldsymbol{x}}_{k+1|k}^{l,(i)} &= \boldsymbol{B}_k^n \boldsymbol{x}_k^{n,(i)} + \boldsymbol{B}_k^l \hat{\boldsymbol{x}}_k^{l,(i)} + \boldsymbol{K}_k (\boldsymbol{x}_{k+1|k}^{n,(i)} - \boldsymbol{A}_k^n \boldsymbol{x}_k^{n,(i)} - \boldsymbol{A}_k^l \hat{\boldsymbol{x}}_k^{l,(i)}) \\ \boldsymbol{P}_{k+1|k} &= \boldsymbol{B}_k^l \boldsymbol{P}_k (\boldsymbol{B}_k^l)^{\mathrm{T}} + \boldsymbol{Q}_k^l - \boldsymbol{K}_k \boldsymbol{M}_k \boldsymbol{K}_k^{\mathrm{T}} \\ \boldsymbol{M}_k &= \boldsymbol{A}_k^l \boldsymbol{P}_k (\boldsymbol{A}_k^l)^{\mathrm{T}} + \boldsymbol{Q}_k^n \\ \boldsymbol{K}_k &= \boldsymbol{B}_k^l \boldsymbol{P}_k (\boldsymbol{A}_k^l)^{\mathrm{T}} \boldsymbol{M}_k^{-1} \end{aligned}$$

步骤 5：重复步骤（2）～步骤（4）。

MPF 算法的复杂度取决于实际的模型和边缘化手段，关于这一点文献[47]已有详细的研究和分析，对于式（9-23）和式（9-24）所示的模型，若非线性状态和线性状态的维数分别为 n 和 l，粒子个数为 N，则算法复杂度为

$$\mathcal{C}(n,l,N)=4nl^2+8ln^2+4n^3/3+5l^3-5nl+2l^2$$
$$+(6nl+4n^2+2l^2+n-l+nc_3+c_1+c_2)N$$
式中：c_1为计算高斯似然的复杂度；c_2为重采样的复杂度；c_3为产生随机数的复杂度。

对于L维状态量$\boldsymbol{x}_k$，根据文献[47]的分析，在相同运算复杂度条件下（即$\mathcal{C}(n+l,0,N_{\text{SPF}})=\mathcal{C}(n,l,N_{\text{MPF}})$）MPF 算法与 SPF 算法所具有的粒子个数之比如图 9.1 所示。

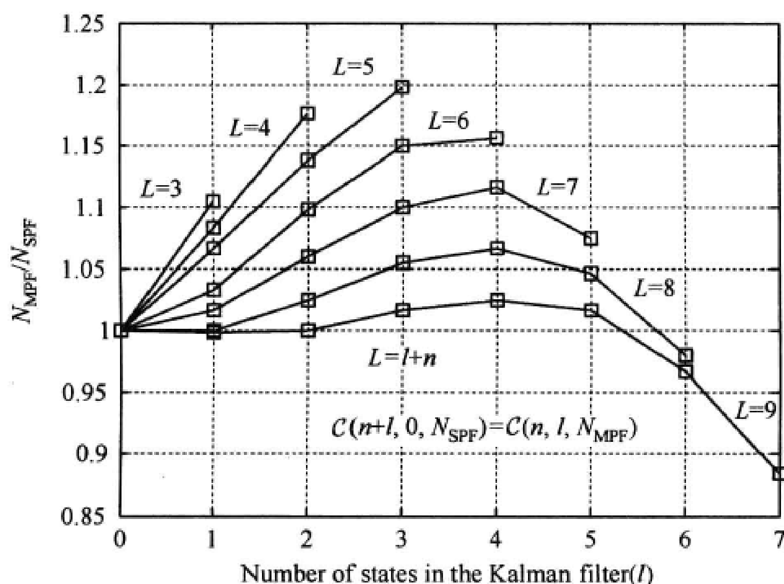


图 9.1 相同运算开销条件下 MPF 与 SPF 粒子个数比

9.3 应用实例

9.3.1 被动雷达目标跟踪

考虑二维平面内的被动目标定位与跟踪问题，目标运动可描述为

$$\boldsymbol{x}_{k+1}=\boldsymbol{F}_k\boldsymbol{x}_k+\boldsymbol{w}_k \tag{9-25}$$

式中：$\boldsymbol{x}_k=[x_k\ y_k\ \dot{x}_k\ \dot{y}_k]^{\text{T}}$为状态矢量；$\boldsymbol{F}_k=\begin{bmatrix}\boldsymbol{I}_2 & T\cdot\boldsymbol{I}_2\\ 0 & \boldsymbol{I}_2\end{bmatrix}$为状态转移矩阵；$T$为观测时间间隔；$\boldsymbol{w}_k$为过程噪声且$\boldsymbol{w}_k\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{Q}_k)$。

观测量包括方位角（φ_k），角度变化率（$\dot{\varphi}_k$）和多普勒频率（f_k），所得非线性观测方程为

$$\boldsymbol{y}_k=\boldsymbol{h}(\boldsymbol{x}_k)+\boldsymbol{v}_k=\begin{bmatrix}\arctan\left(x_k/y_k\right)\\ \dfrac{y_k\dot{x}_k-x_k\dot{y}_k}{r_k^2}\\ f_0\left(1-\dot{r}_k/c\right)\end{bmatrix}+\begin{bmatrix}v_{\varphi_k}\\ v_{\dot{\varphi}_k}\\ v_{f_k}\end{bmatrix} \tag{9-26}$$

式中：$r_k=\sqrt{x_k^2+y_k^2}$；c为光速；f_0为辐射源频率。

显然，上述模型中包含了关于目标速度的线性子结构。若定义$\boldsymbol{x}_k^n=[x_k\ y_k]^{\text{T}}$，$\boldsymbol{x}_k^l=[\dot{x}_k\ \dot{y}_k]^{\text{T}}$，则式（9-25）和式（9-26）所表示的模型可重新记为[48]

$$
\begin{aligned}
\boldsymbol{x}_k^n &= \boldsymbol{x}_{k-1}^n + T\boldsymbol{x}_{k-1}^l + \boldsymbol{w}_{k-1}^n \\
\boldsymbol{x}_k^l &= \qquad\quad \boldsymbol{x}_{k-1}^l + \boldsymbol{w}_{k-1}^l
\end{aligned} \tag{9-27}
$$

$$
\boldsymbol{y}_k = \begin{bmatrix} \varphi_k \\ \dot{\varphi}_k \\ f_k \end{bmatrix} = \underbrace{\begin{bmatrix} \arctan(x_k / y_k) \\ 0 \\ f_0 \end{bmatrix}}_{h(\boldsymbol{x}_k^n)} + \underbrace{\begin{bmatrix} 0 & 0 \\ y_k / r_k^2 & -x_k / r_k^2 \\ -f_0 x_k /(c r_k) & -f_0 y_k /(c r_k) \end{bmatrix}}_{C(\boldsymbol{x}_k^n)} \boldsymbol{x}_k^l + \boldsymbol{v}_k \tag{9-28}
$$

式中

$$
\boldsymbol{w}_k^n \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_k^n), \quad \boldsymbol{w}_k^l \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_k^l), \quad \boldsymbol{v}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_k)
$$

本实例中，首先通过仿真验证 MPF 算法的有效性，随后将其应用于被动目标定位的外场试验数据滤波处理，同时对 SPF 和 MPF 的性能进行比较。

仿真中，假定观测器作折线运动，目标初始位置和速度分别为 $(x_0, y_0) = (30,50)$ km 和 $(\dot{x}_0, \dot{y}_0) = (-150,50)$ m/s，两者航迹如图 9.2 所示。进一步假定 T=1s，f_0=3GHz，$\boldsymbol{Q}_k^n = \boldsymbol{Q}_k^l = 4\boldsymbol{I}_2$，$\boldsymbol{P}_0 = \text{diag}\{1e4, 1e4\}$。方位角、角度变化率和多普勒频率的测量精度分别为 5mrad，0.3mrad/s 和 15Hz。仿真中粒子数目 N=500，蒙特卡罗次数为 100，所得结果如图 9.3 所示。

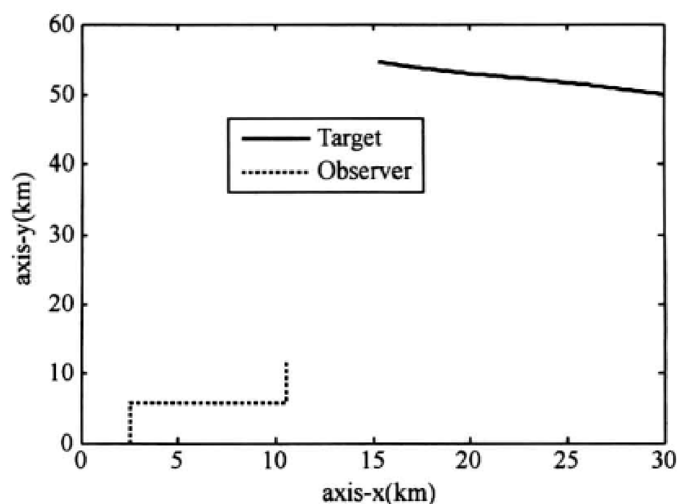


图 9.2　目标与观测器航迹

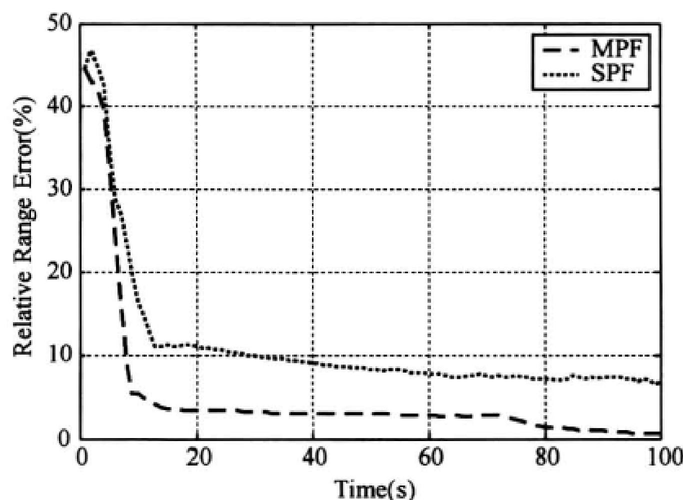


图 9.3　仿真场景中的相对距离误差

由图 9.3 可以看出，在开始一段时间内，SPF 和 MPF 的收敛速度都较快，其相对距离误差在 15s 之内由开始的 45%下降到 10%左右。相比之下，MPF 比 SPF 更加有效，这主要体现在收敛时间和跟踪误差两方面。究其原因，是由于在 MPF 中粒子处于更低维的状态空间，从而在给定算法复杂度的条件下能产生更多有效的粒子来近似真实的 PDF，提高估计性能。

试验场景与第 4 章中的图 4.4 相同，观测器到目标航迹的垂直距离约 520m，辐射源的运动速度约为 15m/s。所不同的是，这里测量参数为方位角、角度变化率和多普勒频率，其测量精度分别为 0.36mrad，2.5mrad/s 和 10Hz。试验中 T=0.5s，N=400，分别用 SPF 和 MPF 对目标进行跟踪，其结果如图 9.4 和图 9.5 所示。

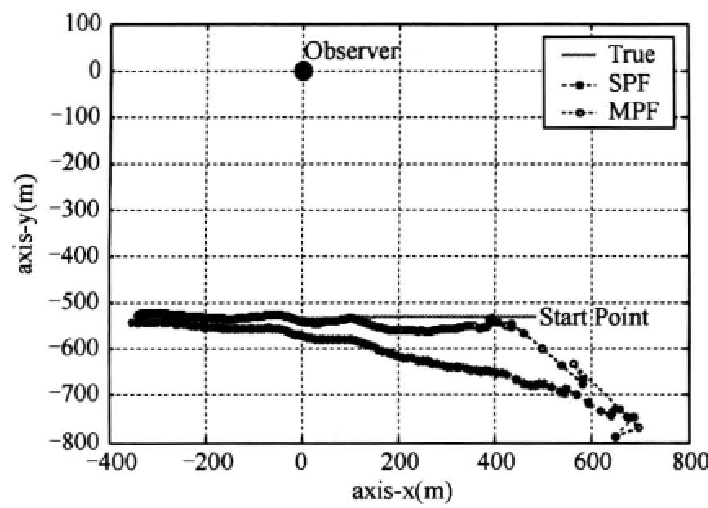


图 9.4 试验场景中的目标航迹跟踪

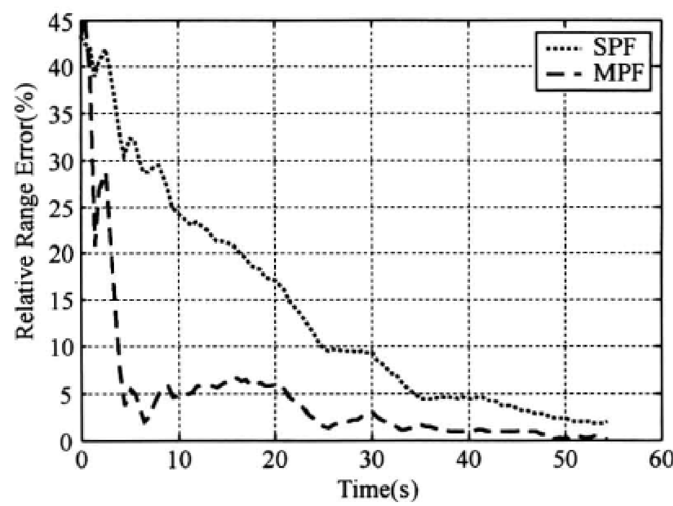


图 9.5 试验场景中的相对距离误差

从图 9.4 和图 9.5 可以清楚地看到，MPF 在外场试验数据处理中取得良好的效果；同时可以看出，MPF 的跟踪性能比 SPF 更好，这一点与前面的理论分析和仿真结果是完全一致的。与数值仿真相比，试验中的跟踪误差更小，这是由观测器和目标的相对位置和距离决定的，在其他条件相同的情况下，两者之间的距离越小，跟踪精度越高，反之亦然。

仿真和试验结果均表明，MPF 在收敛速度方面要优于传统 SPF，是一种较为有效的估计算法；同时该算法可以有效降低估计误差，改善跟踪性能，因此在被动目标跟踪应用中具有潜在的优势。

9.3.2 红外检测前跟踪

考虑二维平面内的目标检测前跟踪问题，目标的运动状态满足

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}\boldsymbol{x}_k + \boldsymbol{w}_k \tag{9-29}$$

式中：$\boldsymbol{x}_k = [x_k\ \dot{x}_k\ y_k\ \dot{y}_k\ I_k]^{\mathrm{T}}$ 为状态矢量；$\boldsymbol{F} = \mathrm{diag}([\boldsymbol{F}_1, \boldsymbol{F}_1, 1])$ 为状态转移矩阵，$\boldsymbol{F}_1=\begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$；$T$ 为观测时间间隔；$\boldsymbol{w}_k$ 为过程噪声且 $\boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{0},\boldsymbol{Q}_k)$。

目标观测方程为

$$z_k^{(i,j)} = \begin{cases} h_k^{(i,j)}(\boldsymbol{x}_k) + v_k^{(i,j)} & H_1 \\ v_k^{(i,j)} & H_0 \end{cases} \tag{9-30}$$

式中：$h_k^{(i,j)}(\boldsymbol{x}_k) = \dfrac{\Delta_x \Delta_y I_k}{2\pi\Sigma^2}\exp\left[-\dfrac{(i\Delta_x - x_k)^2 + (j\Delta_y - y_k)^2}{2\Sigma^2}\right]$，$I_k$ 为信号强度，Σ 为扩散因子；$v_k^{(i,j)}$ 为第 (i,j) 个分辨单元的观测噪声；H_1 和 H_0 代表目标信号的存在性假设，H_1 表示有目标，H_0 表示无目标。

针对上述系统模型结构，若定义 $\boldsymbol{x}_k^n = [x_k\ y_k\ I_k]^{\mathrm{T}}$，$\boldsymbol{x}_k^l = [\dot{x}_k\ \dot{y}_k]^{\mathrm{T}}$，则式（9-29）所表示的模型可重新描述为

$$\begin{aligned} \boldsymbol{x}_k^n &= \boldsymbol{A}\boldsymbol{x}_{k-1}^n + \boldsymbol{B}\boldsymbol{x}_{k-1}^l + \boldsymbol{w}_{k-1}^n \\ \boldsymbol{x}_k^l &= \boldsymbol{C}\boldsymbol{x}_{k-1}^l + \boldsymbol{w}_{k-1}^l \end{aligned} \tag{9-31}$$

式中：

$$\boldsymbol{w}_k^n \sim \mathcal{N}(\boldsymbol{0},\boldsymbol{Q}_k^n),\quad \boldsymbol{w}_k^l \sim \mathcal{N}(\boldsymbol{0},\boldsymbol{Q}_k^l),\quad \boldsymbol{A}=\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},\quad \boldsymbol{B}=\begin{bmatrix} T & 0 \\ 0 & T \\ 0 & 0 \end{bmatrix},\quad \boldsymbol{C}=\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

此时，观测方程可进一步记为

$$z_k^{(i,j)} = \begin{cases} h_k^{(i,j)}(\boldsymbol{x}_k^n) + v_k^{(i,j)} & H_1 \\ v_k^{(i,j)} & H_0 \end{cases} \tag{9-32}$$

式中：$v_k^{(i,j)} \sim \mathcal{N}(0,\sigma^2)$；$\sigma^2$ 为噪声功率。

由此可见，上述模型满足 9.2.3 节中介绍的系统模型形式，可采用边缘化粒子滤波算法进行处理。不过，与单纯的目标跟踪不同，对于检测前跟踪问题，不仅要实现对目标运动状态的估计，同时还要对目标的存在性进行判断（即进行目标检测）。为此，通常需要引入一独立变量 $E_k \in \{0,1\}$ 来表示目标的出现与否，其中 0 表示目标未出现，而 1 表示目标出现。

若定义目标“出现”概率 P_b 和“消失”概率 P_d 分别为

$$P_b \triangleq P\{E_k = 1 \mid E_{k-1} = 0\} \tag{9-33}$$

$$P_d \triangleq P\{E_k = 0 \mid E_{k-1} = 1\} \tag{9-34}$$

同时假定目标持续出现概率（存活概率 $P\{E_k = 1 \mid E_{k-1} = 1\}$）和目标持续不出现概率 $P\{E_k = 0 \mid E_{k-1} = 0\}$ 分别为 $1-P_d$ 和 $1-P_b$，则可得到关于目标存在性的转移概率矩阵为

$$\boldsymbol{\Pi} = \begin{bmatrix} 1-P_b & P_b \\ P_d & 1-P_d \end{bmatrix} \tag{9-35}$$

在具体应用中，一般假设目标在初始时刻的出现概率 $P(E_0=1)$ 已知，用于表征目标出现与否的变量 E_k 通常可利用以下两种途径得到。

第一种为混合状态估计，在目标状态向量中增加该离散变量，实现目标状态与 E_k 的混合估计过程，即估计密度 $p(\boldsymbol{x}_k, E_k=1 \mid \boldsymbol{z}_{1:k})$。$k$ 时刻的独立变量 E_k 基于转移概率矩阵 $\boldsymbol{\Pi}$ 和 $k-1$ 时刻的独立变量 E_{k-1} 计算得到，目标状态 $\boldsymbol{x}_k$ 由粒子滤波处理获取。完成滤波后，计算 k 时刻表示目标出现（$E_k=1$）的粒子数量（N_e）与粒子总数（N）的比值，将该比值与预定义门限（P_{th}）进行比较，超过门限即认为检测到目标；反之，则判定目标没有出现。

第二种算法仍然是估计密度 $p(\boldsymbol{x}_k, E_k=1 \mid \boldsymbol{z}_{1:k})$，但是处理中将独立变量 E_k 的估计从目标状态向量中分离出来，采取更为有效的出现概率计算方法，具体处理过程这里不再详述，可参见文献[49]和文献[50]。分离的结果使得对于粒子的使用更为有效，这是因为所有的粒子都用于表示目标“出现”，而没有表示目标“消失”的粒子。

另外，在检测前跟踪应用中，计算粒子权重时采用的是似然比形式而非传统滤波应用中的似然形式，即

$$l(z_k^{(i,j)}|\boldsymbol{x}_k^n) \triangleq \frac{p_{S+N}(z_k^{(i,j)}|\boldsymbol{x}_k^n)}{p_N(z_k^{(i,j)})} = \exp\left\{-\frac{h_k^{(i,j)}(\boldsymbol{x}_k^n)[h_k^{(i,j)}(\boldsymbol{x}_k^n) - 2z_k^{(i,j)}]}{2\sigma^2}\right\} \tag{9-36}$$

于是，对于第 $s(s=1,2,\cdots,N)$ 个粒子，其权值可表示为

$$w_k^{(s)} = \begin{cases} \prod\limits_{i \in C_i(\boldsymbol{x}_k^{n,(s)})} \prod\limits_{j \in C_j(\boldsymbol{x}_k^{n,(s)})} l(z_k^{(i,j)} \mid \boldsymbol{x}_k^{n,(s)}) & E_k^s=1 \\ 1 & E_k^s=0 \end{cases} \tag{9-37}$$

完整的检测前跟踪算法流程如下。

Alg（9.3）：MPF based Track-before-Detect

步骤 1：初始化

在 $k=0$ 时刻，按照下面的伪码获取初始粒子集 $\{\boldsymbol{x}_0^{n,(s)}\}_{s=1}^N$ 及 $\{\boldsymbol{x}_0^l, \boldsymbol{P}_0\}$：

•for s=1:N

$-x_0^{n,(s)} \sim \mathcal{U}[0,n_x]$, $y_0^{n,(s)} \sim \mathcal{U}[0,n_y]$

$I_0^{(n)} \sim \mathcal{U}[I_{\min}, I_{\max}]$

$\boldsymbol{x}_0^{n,(s)} = [x_0^{n,(s)} \ y_0^{n,(s)} \ I_0^{(n)}]^{\mathrm{T}}$

•end for

式中：$\mathcal{U}$ 表示均匀分布；$I_{\min}$ 和 $I_{\max}$ 表示目标信号强度的可能范围。同时根据目标存在的先验概率 $P(E_0=1)$ 随机地选择一部分粒子使其状态置为空，即 $\boldsymbol{x}_0^{n,(s)} = \varnothing$。

对于 $k \geqslant 1$，采用以下步骤递推。

步骤 2：权值计算

通过式（9-36）和式（9-37）计算权重 $w_k^{(s)}$ 并归一化得到 $\tilde{w}_k^{(s)}$。

步骤 3：测量更新

对非线性部分，通过粒子重采样使得 $\Pr(\boldsymbol{x}_k^{n,(i)} = \boldsymbol{x}_{k|k-1}^{n,(s)}) = \tilde{w}_k^{(s)}$，并得到新的粒子集为 $\{\boldsymbol{x}_k^{n,(i)}\}_{i=1}^N$；对线性部分直接通过上一时刻的估计获得更新，即 $\hat{\boldsymbol{x}}_k^l = \hat{\boldsymbol{x}}_{k|k-1}^l$，$\boldsymbol{P}_k = \boldsymbol{P}_{k|k-1}$。

步骤 4：时间更新

非线性部分通过以下算式预测新的粒子：

$$\boldsymbol{x}_{k+1|k}^{n,(i)} \sim \mathcal{N}\left(\boldsymbol{x}_{k+1}^{n};\boldsymbol{A}\boldsymbol{x}_{k}^{n,(i)}+\boldsymbol{B}\hat{\boldsymbol{x}}_{k}^{l},\boldsymbol{Q}_{k}^{n}+\boldsymbol{A}_{k}^{l}\boldsymbol{P}_{k}(\boldsymbol{A}_{k}^{l})^{\mathrm{T}}\right)$$

线性部分则通过卡尔曼滤波公式计算：

$$\hat{\boldsymbol{x}}_{k+1|k}^{l,(i)}=\boldsymbol{B}\hat{\boldsymbol{x}}_{k}^{l,(i)}+\boldsymbol{K}_{k}(\boldsymbol{x}_{k+1|k}^{n,(i)}-\boldsymbol{A}\boldsymbol{x}_{k}^{n,(i)}-\boldsymbol{B}\hat{\boldsymbol{x}}_{k}^{l,(i)})$$

$$\boldsymbol{P}_{k+1|k}=\boldsymbol{C}\boldsymbol{P}_{k}\boldsymbol{C}^{\mathrm{T}}+\boldsymbol{Q}_{k}^{l}-\boldsymbol{K}_{k}\boldsymbol{M}_{k}\boldsymbol{K}_{k}^{\mathrm{T}}$$

$$\boldsymbol{M}_{k}=\boldsymbol{B}\boldsymbol{P}_{k}\boldsymbol{B}^{\mathrm{T}}+\boldsymbol{Q}_{k}^{n}$$

$$\boldsymbol{K}_{k}=\boldsymbol{C}\boldsymbol{P}_{k}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{M}_{k}^{-1}$$

步骤 5：检测概率计算

统计代表目标存在（$E_k^{(s)}=1$）的粒子数 N_e，由此得到检测概率的估值 $\hat{P}_D=N_e/N$；若该值大于门限 P_{th} 表示检测到目标，则进入步骤 6；反之，直接转到步骤 7。

步骤 6：状态估计

$$\hat{\boldsymbol{x}}_{k}^{n}=\sum_{i=1}^{N}E_{k}^{i}\boldsymbol{x}_{k}^{n,(i)}/\sum_{i=1}^{N}E_{k}^{i}\text{，}\quad \hat{\boldsymbol{x}}_{k}^{l}=\sum_{i=1}^{N}E_{k}^{i}\boldsymbol{x}_{k}^{l,(i)}/\sum_{i=1}^{N}E_{k}^{i}$$

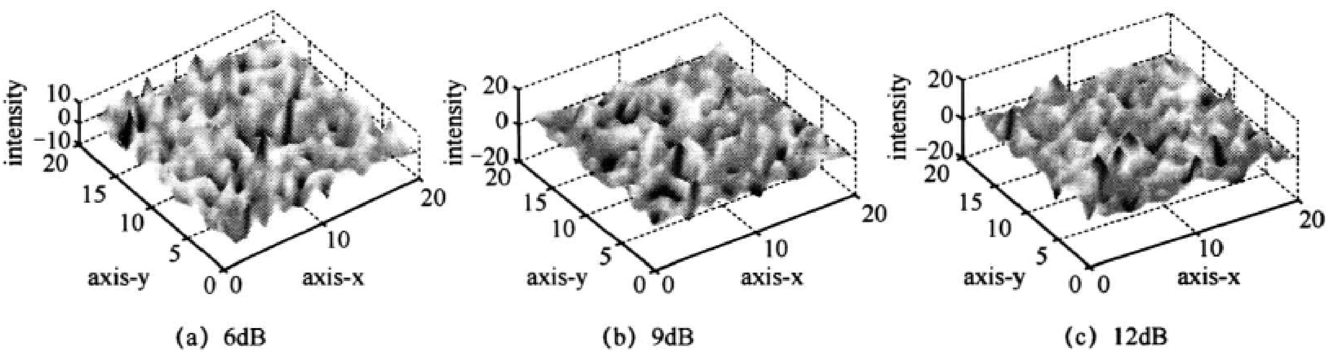
步骤 7：令 $k=k+1$，重复步骤 2～步骤 5

仿真中，假定目标初始位置为 (4.5,7.2)，初始速度为 (0.25,0.1)，初始时刻的信号强度 I_0 根据信噪比 SNR 来确定，$\boldsymbol{Q}_k^n$=1e-4$\boldsymbol{I}_3$，$\boldsymbol{Q}_k^l$=1e-4$\boldsymbol{I}_2$，其他参数设置见表 9.1。

表 9.1　MPF-TbD 滤波算法参数设置

变量名	n_x,n_y	Δ_x,Δ_y	P_b,P_d	σ	Σ	P_{th}	T
参数值	20	1	0.05	3	0.7	0.6	1

设总共进行 40 个采样时间的观测（产生 40 帧数据），开始时监视区内无目标，第 k=6 时刻目标开始出现并持续到 k=35 时刻，之后目标消失。在粒子数目 N=1000，信噪比 SNR=6dB,9dB,12dB 条件下，分别进行 200 次实验，所得结果如图 9.6～图 9.8 所示。



(a) 6dB　(b) 9dB　(c) 12dB

图 9.6　不同信噪比条件下的图像

图 9.6 给出的是不同信噪比条件下的第 15 帧图像，可见通过肉眼很难看出是否存在目标，更无法确定目标位于哪个位置，图 9.7 是通过边缘化粒子滤波得到的目标检测结果。

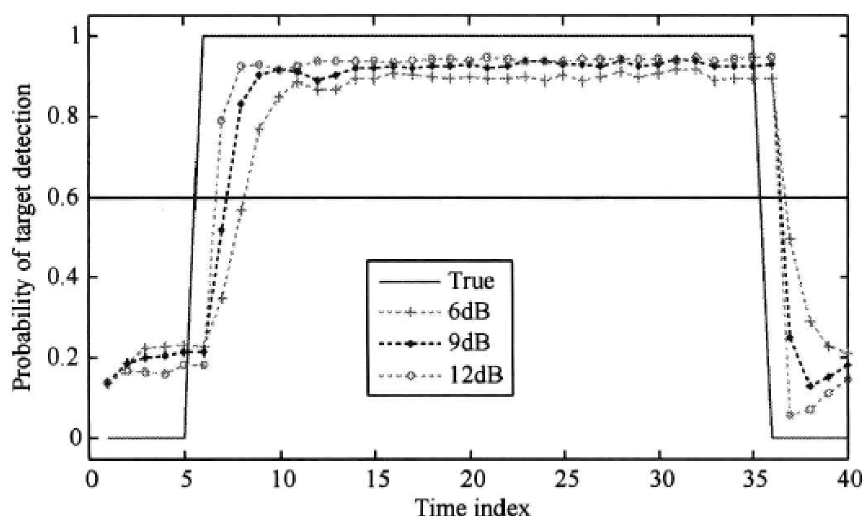


图 9.7　不同信噪比条件下的检测概率

由图 9.7 中的结果可以看出，随着信噪比的提高，目标检测概率也有不同程度的提高；当信噪比大于 6dB 时，对目标出现的检测有 2 个时刻的时延，而当信噪比为 6dB 时的检测时延为 3 个时刻；在不同信噪比条件下，对目标消失的检测均存在 1 个时刻的时延。

图 9.8 和图 9.9 进一步给出了不同信噪比下的目标位置和速度估计 ARMSE，为了评价的客观性，只对检测到目标的情况下（时刻 9～时刻 35）进行统计。由图中的结果可见，随着信噪比的提高，检测前跟踪的状态估计精度也进一步提高；不过，当信噪比大到一定程度后（如大于 14dB），信噪比的提高对估计性能的改善不再明显（说明在高信噪比条件下影响 TbD 性能的主要因素已不再是检测结果）。

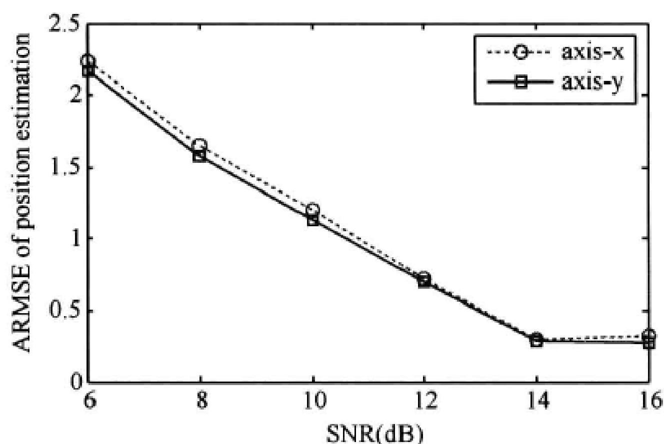


图 9.8　位置估计平均均方根误差

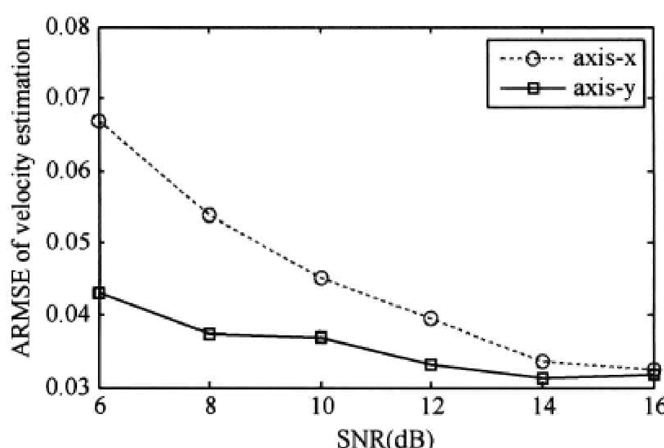


图 9.9　速度估计平均均方根误差

由此可见，采用边缘化粒子滤波技术可实现对低信噪比条件下目标检测前跟踪。当然，上述例子给出的只是一般的实现过程，实际中为了得到更好的检测前跟踪性能，可能需要辅以其他的手段，如对非线性部分进行粒子更新时可融入观测量信息，并在检测环节增加特殊的策略如引入序贯检测[50]等以提高对低信噪比条件的适应能力。

9.4 本章小结

尽管粒子滤波算法在处理非线性问题中具有普遍的适用性，但其估计性能依赖于粒子对状态后验分布的近似程度。在给定的估计性能要求下，算法所需的粒子数随待估状态维数的增加呈指数增长，使得这一技术在很多应用中受到限制。但在实际应用中，很多非线性系统模型都具有线性的子结构，在这种条件下采用“边缘化”技术将模型分解成线性和非线性两个子结构，使粒子滤波器只工作在低维状态空间，可提高滤波算法的效费比。

本章主要阐述“边缘化”粒子滤波的算法原理，并给出了几种典型系统模型的算法实现过程，在此基础上结合两个不同的目标跟踪实例介绍了边缘化粒子滤波的具体应用。需要指出的是，在相同的粒子数目条件下，MPF 的运算量要低于传统基于 SIS 技术的粒子滤波器（如前面讨论的 EPF、UPF 等），但通常要高于 SPF 的运算开销。因此，在实际应用中，应根据具体的问题形式及估计性能指标要求等，综合分析不同算法的效费比，从而完成对算法的合理选择。

第 10 章　智能优化采样滤波

第 7 章中指出，标准粒子滤波算法尽管在实现上较为方便，但由于仅利用了先验分布信息，存在严重的粒子退化现象及近似效率低下问题。第 8 章从不同的角度阐述了基于建议分布函数的粒子采样方法，以此来提高滤波器的估计性能。在这类方法中，主要通过融入量测信息提高粒子的产生质量，从而改善对状态后验分布的近似效果。

事实上，导致标准粒子算法效率低下的主要原因之一是粒子退化问题，若能采用某种手段减轻或消除这一现象，则可以起到提高粒子滤波估计性能的目的。遗传算法是模拟自然界生物进化过程与机制的一种随机优化与搜索方法，它反映的是一种进化思想，这种思想为解决粒子退化问题提供了一条重要思路。由于遗传算法本质上是一种智能优化方法，推而广之，其他智能优化方法如人工免疫算法、人工鱼群算法及粒子群优化算法等也具备这种潜能。

本章主要阐述基于智能优化采样的粒子滤波算法，重点介绍遗传算法、免疫算法和人工鱼群算法的基本原理及与粒子滤波框架的结合方法，分别得到遗传粒子滤波（GPF）[52-57]、人工免疫粒子滤波（AIPF）[58-60]和人工鱼群粒子滤波（AFSPF）[61-62]算法。在 GPF 算法中，主要是通过选择、交叉、变异等操作代替传统的重采样方法，依此防止粒子的退化现象；在 AIPF 算法中，将粒子看作是抗体，应用免疫算法的优化机理在粒子更新过程中对每个抗体进行克隆，从而得到优选的抗体，保持样本集的多样性；而在 AFSPF 算法中，将人工鱼群的个体视为粒子，通过将觅食、聚群行为与粒子更新相结合，使得先验粒子向高似然区域移动，从而改善粒子的分布。

10.1　遗传粒子滤波

遗传算法（GA）是模拟自然界生物进化过程与机制来求解最优化问题的一类自组织、自适应人工智能技术。它通过模拟达尔文“优胜劣汰、适者生存”的原理激励好的结构，同时寻找最优结构，是一种随机的优化与搜索方法。遗传算法反映的是一种进化思想，而粒子滤波器中存在的主要问题是粒子退化，用遗传进化的思想来改善粒子的分布特性，可有效提高粒子滤波的估计效果。

10.1.1　遗传算法基本原理

GA 是一种基于自然选择和自然遗传原理的随机搜索技术，通过选择、交叉和变异实现种群的更新和优化。遗传算法每次迭代都保存一个最优种群，而不仅仅是一个最优个体。在给出具体的算法前，首先介绍一下 GA 涉及的一些重要概念，见表 10.1。

表 10.1　遗传算法基本术语表

术　语	含　义
遗传因子 gene	也称为基因，是构成染色体的基本单位，即个体编码中的每一位
染色体 chromosome	生物细胞中含有一种微小的丝状化合物，它是遗传物质的主要载体，由多个遗传因子—基因组成
个体 population	组成种群的解空间中的一个解。遗传算法的中心思想就是通过遗传操作从母体中找出优良的个体并使优良的状态遗传保持到下一代
种群 Group	解空间的一组作为父代遗传用的初始解
编码 Coding	遗传操作通常不能直接处理数值变量，只能针对编码串进行，这样，就需要将数值变量转化为一个二进制编码串，这个步骤就是编码
解码 Decoding	解码是编码的逆过程，完成从一个基因串还原出真实数值变量的过程
适应度 Fitness	在研究自然界中生物的遗传和进化现象时，生物学家使用适应度这个术语来度量某个物种对于生存环境的适应程度。在遗传算法中，用适应度来评价个体优劣程度
选择 Selection	指决定以一定的概率从种群中选择若干个体的操作。一般而言，选择的过程就是基于适应度的优胜劣汰过程，某个体被选择的概率与其适应度值成正比
交叉 Crossover	从种群中随机选取两个个体做父体和母体，再依交叉概率随机地选取一交叉位，然后将两个体中位于交叉位后的符号串互换，保留交叉位前的各基因位符号不变，形成两个新个体的遗传操作，这个过程又称为基因重组，俗称"杂交"，它是获取优良个体的重要手段
变异 Mutation	对个体编码中的每一位基因位根据变异概率进行 0、1 翻转，形成新个体的遗传操作。变异操作可使适应值小的个体发生变异或在整个群体性态趋于一致时迫使个体发生变化，同时，防止适应值大的个体变异，从而使每一代种群保持新鲜个体，避免种群进化停滞而过早收敛

在 GA 的运算过程中，通常不对所求解问题的实际决策变量直接进行操作，而是对表示可行解的个体编码施加选择、交叉、变异等遗传运算。因此，GA 一般都离不开编码和解码问题，常用的编码方法是二进制编码方法。同时，针对某个具体问题，编码和解码操作也会增加算法实现的复杂度。若能直接针对实数进行遗传操作、避免编解码过程，则可有效降低算法实现的复杂度。

GA 是一个迭代过程，同时也是一个概率搜索算法。假设第 k 代种群为 $G_k=\{P_k^{(1)}, P_k^{(2)},\cdots,P_k^{(N)}\}$，其中，$N$ 表示该种群中的个体数（也即染色体数），则利用迭代方法，通过选择、交叉、变异操作就能得到 $k+1$ 代种群 $G_{k+1}=\{P_{k+1}^{(1)},P_{k+1}^{(2)},\cdots,P_{k+1}^{(N)}\}$。具体过程可概括为以下三个步骤。

步骤 1：$k=0$

初始化：随机产生一组初始解，作为初始种群 $P_{k,\{N\}}=\{P_k^{(i)},i=1,2,\cdots,N\}$。

个体评价：计算种群 $P_{k,\{N\}}$ 中每一个体的适应度。

步骤 2：$k=k+1$

选择操作：对种群 $P_{k,\{N\}}$ 进行选择得到种群 $P_{k+1,\{N\}}$。

交叉操作：对种群 $P_{k+1,\{N\}}$ 中的个体进行交叉运算。

变异操作：对种群 $P_{k+1,\{N\}}$ 中的个体进行变异运算。

种群评估：评估计算出的种群 $\tilde{P}_{k+1,\{N\}}$。

步骤 3：终止条件判断

若进化代数 k（迭代次数）小于设定值或代价函数大于某一门限值，转到步骤 2。

在上述几大步骤中，遗传进化运算的具体实现方法如下：

（1）初始化 $P_{0,\{N\}}$。由先验概率分布 $p(P_0)$ 随机得到初始化群体 $P_{0,\{N\}}=\{P_0^{(i)}\}_{i=1}^N$，即 $P_0^{(i)} \sim p(P_0)$。

（2）个体评价 $P_k^{(i)}$。个体评价即计算该种群中每一个体（染色体）的适应度 $w_k^{(i)}$，可以通过适应度函数 $fit(P_k^{(i)})$ 来计算，该函数通常可根据实际需要来定义，如 $w_k^{(i)} = fit(P_k^{(i)}) = \left|y - y(P_k^{(i)})\right|$。因此，种群中个体的适应度为 $w_k^{(i)} = fit(P_k^{(i)})(i=1,2,\cdots,N)$。

（3）选择操作。新种群 $P_{k,\{N\}}$ 可以按照选择概率从上一代种群 $P_{k-1,\{N\}}$ 中选择得到，有很多选择方法，转盘操作方法就是其中的一种。

利用适应度函数，可得每一个体的概率 $\tilde{w}_{k-1}^{(i)}$ 为

$$\tilde{w}_{k-1}^{(i)} = w_{k-1}^{(i)} / \sum_{j=1}^{N} w_{k-1}^{(j)},\quad i=1,2,\cdots,N$$

由此得到每一个体 $P_{k-1}^{(i)}$ 的累积概率 $c_{k-1}^{(i)}$ 为

$$c_{k-1}^{(i)} = \sum_{j=1}^{i} \tilde{w}_{k=1}^{(j)},\quad i=1,2,\cdots,N$$

在区间 $[0,1]$ 内随机产生一个随机数，即 $u\sim\mathcal{U}[0,1]$，转动转盘，按照下列规则选择一个个体作为下一代种群中的新个体：

若 $u < c_{k-1}^{(i)}$，选择第 1 个个体 $P_{k-1}^{(1)}$，并令 $P_k^{(i)} = P_{k-1}^{(1)}$；

若 $u \geqslant c_{k-1}^{(i)}$，选择第 j 个个体，且 $c_{k-1}^{(j-1)} < u \leqslant c_{k-1}^{(j)}$，令 $P_k^{(i)} = P_{k-1}^{(j)}$，$i=1,2,\cdots,N$。

（4）交叉操作。设交叉概率为 P_c，产生随机数 $u\sim\mathcal{U}[0,1]$。若 $u < P_c$，从种群 $P_{k,\{N\}}$ 中随机选择两个个体 $\{P_k^{(m)}, P_k^{(n)}\}$，按照下式进行配对，并用新个体 $\{\widetilde{P}_k^{(m)}, \widetilde{P}_k^{(n)}\}$ 代替旧个体 $\{P_k^{(m)}, P_k^{(n)}\}$。

$$\widetilde{P}_k^{(m)} = \alpha P_k^{(m)} + (1-\alpha) P_k^{(n)}$$
$$\widetilde{P}_k^{(n)} = \alpha P_k^{(n)} + (1-\alpha) P_k^{(m)}$$

式中：$\alpha \in [0,1]$，其大小可根据具体问题而定。

（5）变异操作。假设变异概率为 P_m，产生随机数 $u\sim\mathcal{U}[0,1]$。若 $u < P_m$，从种群 $P_{k,\{N\}}$ 中随机选择一个个体 $P_k^{(j)}$，按照下式进行变异：

$$\widetilde{P}_k^{(j)} = P_k^{(j)} + \boldsymbol{\eta}$$

式中

$$\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \Sigma)$$

由上面的分析中可以看出，交叉、变异等操作都是直接对实数进行的，这样可避免传统遗传方法中的二进制编码和解码操作环节，因此可以根据实际需要自定义交叉、变异操作算子，更方便于算法的实现。另外，仔细分析不难发现，GA 算法与前文所述的

粒子滤波算法无论在模型还是运算过程中都具有明显的相似性，两者之间的类比关系如表 10.2 和图 10.1 所示。

表 10.2 粒子滤波算法与遗传算法模型比较

粒子滤波算法	遗 传 算 法	粒子滤波算法	遗 传 算 法
粒子 $\boldsymbol{x}_k^{(i)}$	个体 $P_k^{(i)}$	过程噪声	交叉、变异
粒子集 $\{\boldsymbol{x}_k^{(i)}\}_{i=1}^N$	种群 $\{P_k^{(i)}\}_{i=1}^N$	重要性权值 $w_k^{(i)}$	个体的适应度 $fit(P_k^{(i)})$
滤波	选择	观测值 $\boldsymbol{y}_k$	遗传环境（目标函数）

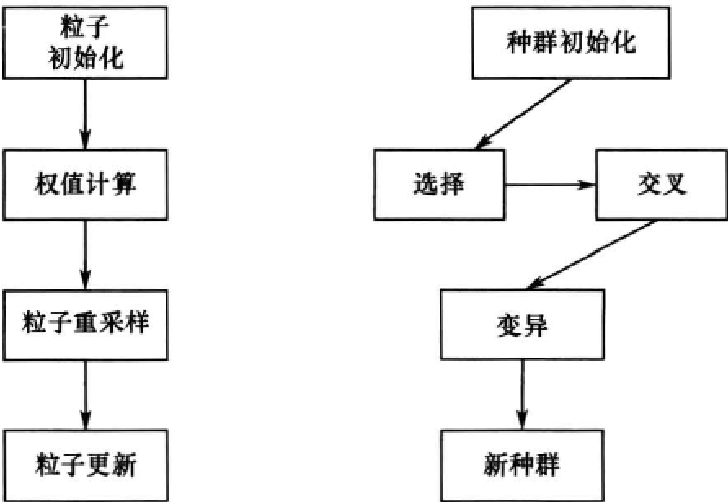


图 10.1 粒子滤波与遗传算法执行过程类比

10.1.2 遗传粒子滤波算法

遗传粒子滤波（GPF）算法主要是将 GA 中的选择、交叉、变异等进化思想引入到粒子滤波算法中，代替传统的重采样方法。由于 GA 具有独特的寻优能力，可以提高粒子的使用效率，使逼近后验概率分布最大值所需的粒子数大大减少；同时，由于 GA 能有效增加粒子的多样性，可有效防止出现滤波发散现象，提高状态估计精度。

GPF 算法中的选择、交叉、变异操作实现步骤描述如下。

（1）选择操作。对 k 时刻粒子集 $\{\boldsymbol{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^N$， $w_k^{(i)}$ 可视为此粒子集中对应粒子的适应度，适应度的方差对应粒子重要性权值的方差。由于粒子重要性权值的方差最小时，状态估计最接近真实状态。因此，根据适应度的方差大小来判断选择操作。若方差大小符合要求，进行选择操作，保留所有粒子，而不再进行交叉、变异操作，直接进入预测阶段；若方差大小不符合要求，跳过选择操作，直接进行交叉、变异操作。

（2）交叉操作。从粒子集中随机选取两个粒子 $\{\boldsymbol{x}_k^{(m)}, \boldsymbol{x}_k^{(n)}\}_{m,n=1}^N$，按照下式进行交叉操作。

$$\tilde{\boldsymbol{x}}_k^{(m)} = \alpha \boldsymbol{x}_k^{(m)} + (1-\alpha)\boldsymbol{x}_k^{(n)} + \boldsymbol{\eta}$$
$$\tilde{\boldsymbol{x}}_k^{(n)} = \alpha \boldsymbol{x}_k^{(n)} + (1-\alpha)\boldsymbol{x}_k^{(m)} + \boldsymbol{\eta}$$

式中： $\boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{x};\boldsymbol{0},\Sigma)$ 为正态分布； $\alpha \sim \mathcal{U}(0,1)$ 为均匀分布。

（3）变异操作。从粒子集中随机选择一个粒子 $\{\boldsymbol{x}_k^{(j)}\}_{j=1}^N$，按照下式进行变异操作：

$\tilde{\boldsymbol{x}}_k^{(j)} = \boldsymbol{x}_k^{(j)} + \boldsymbol{\eta}$

变异规则为：如果 $p(z_k|\tilde{x}_k^{(j)})>p(z_k|x_k^{(j)})$，则接受粒子 $\tilde{x}_k^{(j)}$；否则，产生 $u\sim\mathcal{U}(0,1)$，并判断 $u\leqslant p(z_k|\tilde{x}_k^{(j)})/p(z_k|x_k^{(j)})$ 是否成立，成立时接受粒子 $x_k^{(j)}$，不成立则接受粒子 $\tilde{x}_k^{(j)}$。

对于式（10-1）和式（10-2）所描述的非线性系统

$$\boldsymbol{x}_{k+1}=f(\boldsymbol{x}_k)+\boldsymbol{w}_k \tag{10-1}$$

$$\boldsymbol{z}_k=h(\boldsymbol{x}_k)+\boldsymbol{v}_k \tag{10-2}$$

结合选择、交叉和变异的操作过程，可将 GPF 算法总结如下。

Alg（10.1）：Genetic Particle Filtering

步骤 1：初始化

由先验概率 $p(\boldsymbol{x}_0)$ 产生粒子集 $\{\boldsymbol{x}_0^{(i)}\}_{i=1}^N$，并令所有粒子权值为 $1/N$。

步骤 2：权值更新

在 k 时刻，计算粒子权值

$$w_k^{(i)}=w_{k-1}^{(i)}\frac{p(\boldsymbol{z}_k\left|\boldsymbol{x}_k^{(i)}\right.)p(\boldsymbol{x}_k^{(i)}\left|\boldsymbol{x}_{k-1}^{(i)}\right.)}{q(\boldsymbol{x}_k^{(i)}\left|\boldsymbol{x}_{k-1}^{(i)},\boldsymbol{z}_k\right.)},\quad i=1,2,\cdots,N$$

并且归一化

$$\tilde{w}_k^{(i)}=w_k^{(i)}/\sum_{i=1}^N w_k^{(i)}$$

步骤 3：遗传操作

（1）选择操作。计算粒子权值方差，若方差符合要求，保留粒子为 $\{\tilde{\boldsymbol{x}}_k^{(i)},\tilde{w}_k^{(i)}\}_{i=1}^N$，转到步骤 4；否则，直接进行下一步操作。

（2）采用前文所述的方法进行粒子的交叉、变异操作，从而得到新的粒子集 $\{\tilde{\boldsymbol{x}}_k^{(i)},\tilde{w}_k^{(i)}\}_{i=1}^N$。

步骤 4：估计输出

状态估计：$\hat{\boldsymbol{x}}_k=\sum_{i=1}^{N_s}\tilde{w}_k^{(i)}\tilde{\boldsymbol{x}}_k^{(i)}$

协方差估计：$\boldsymbol{P}_k=\sum_{i=1}^{N_s}\tilde{w}_k^{(i)}(\tilde{\boldsymbol{x}}_k^{(i)}-\hat{\boldsymbol{x}}_k)(\tilde{\boldsymbol{x}}_k^{(i)}-\hat{\boldsymbol{x}}_k)^{\mathrm{T}}$

步骤 5：状态预测

利用状态方程 $f(\cdot)$ 预测下一时刻的粒子 $\boldsymbol{x}_{k+1}^{(i)}$，即

$$\boldsymbol{x}_{k+1}^{(i)}=f(\tilde{\boldsymbol{x}}_k^{(i)})+\boldsymbol{w}_k,\quad i=1,2,\cdots N$$

步骤 6：时刻 $k=k+1$，转到步骤 2

需要说明的是，在步骤 2 中，$q(\boldsymbol{x}_k^{(i)}\left|\boldsymbol{x}_{k-1}^{(i)},\boldsymbol{z}_k\right.)$ 为产生粒子的建议分布，通常可选择为状态的先验分布即 $p(\boldsymbol{x}_k^{(i)}\left|\boldsymbol{x}_{k-1}^{(i)}\right.)$。

10.2 人工免疫粒子滤波

免疫优化是受生物免疫机制启发而形成的一种新型的算法，它模拟了生物免疫系统产生抗体抵御外来入侵抗原的自适应能力，可实现入侵抗原识别、多样抗体生成、

自我调节、免疫记忆等功能。生物免疫系统虽然复杂，但是其呈现出来的抵御抗原的自适应能力却是十分明显的。人工免疫算法是一种基于人工免疫系统的随机全局搜索算法，具有全局多样性寻优能力。将人工免疫算法的寻优能力与粒子滤波算法相结合，可从众多粒子中精选出较好的粒子，改善样本集的多样性，从而提高粒子滤波的估计能力。

10.2.1 人工免疫算法原理

免疫优化算法通过模拟免疫系统的抗体多样性保持机制，实现免疫系统的自我调节功能，即利用抗体的浓度选择机制，实现对抗体的促进和抑制作用，保持抗体的多样性。为了更好地理解免疫算法的基本原理，首先对几个重要概念作一简要介绍。

1. *B 细胞*

B 细胞是 B 淋巴细胞的简称。B 细胞是体内唯一能够产生抗体（免疫球蛋白分子）的细胞，其表面含有可以识别特异性抗原的多种抗体分子，多样性来自千百万种不同的 B 细胞克隆。每一个 B 细胞克隆的特性是由其遗传性决定的，可产生一种能与抗原特异相结合的免疫球蛋白分子。B 细胞有三个主要功能：产生抗体、提呈抗原和分泌细胞因子参与免疫调节。

2. *抗原*

在现代免疫学中，抗原（Antigen，Ag）是指能够诱导免疫系统发生免疫应答并能与其产生的抗体或效应细胞在体内或体外发生特异性反应的物质，抗原表面被抗体识别的部分称为抗原决定基。抗原对应免疫算法中的目标函数。

3. *抗体*

抗体（Antibody，Ab）是 B 细胞识别抗原后克隆扩增分化为浆细胞所产生的一种蛋白质分子，也称为免疫球蛋白分子。抗体结合由外部入侵的感染微组织或者有毒物等抗原，然后依靠自己或者借助免疫系统其他元素帮助破坏这些抗原，消除对人体的威胁。抗体由抗体决定基和独特型组成，抗体决定基是抗体上识别抗原决定基的部分，而独特型是抗体上可供自身免疫细胞识别的抗原决定基。抗体对应目标函数的输入变量。

4. *亲和度*

免疫系统中的免疫识别是发生在分子水平的，并且这种识别是基于抗体决定基和抗原决定基之间的形状互补。发生免疫识别的抗体决定基和抗原决定基在结构上越互补，结合就越可能发生，结合的力度也就越紧密，这种结合的力度与强度被称为抗体与抗原之间的亲和度（affinity）。亲和度对应最优化问题中的函数代价/测度值。

5. *亲和度成熟*

随着抗体分子与同种抗原的多次遭遇，记忆应答中的抗体与较早的初始应答中的抗体相比具有更高的平均亲和度，这个现象称为免疫应答的亲和度成熟。由此可见，免疫应答的亲和度成熟要求成熟应答中的抗体分子的抗原结合点在结构上与在初次应答中抗体分子的抗原结合点不同。

人工免疫算法适合解决目标优化任务，尤其是多峰值目标优化问题，其原理框图如图 10.2 所示。

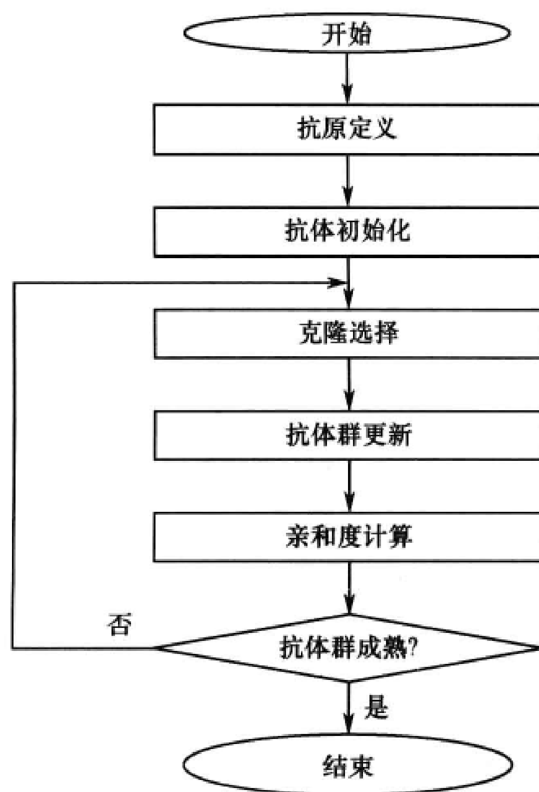


图 10.2　免疫算法原理框图

根据上述原理图，可得人工免疫算法（AIA）的具体流程如下。

Alg（10.2）：Artificial Immune

步骤 1：建立目标函数 $f(\cdot)$，即引入抗原。

由于多目标优化问题最终可以转化为一个待优化（最大化或最小化）的目标函数 $f(\cdot)$，故在抗原种群 Ag 中只包含一个抗原。

步骤 2：令 k=0（k 为种群的代数），随机产生 N 个初始抗体，构成初始抗体种群 $Ab_{k,\{N\}}=\{Ab_k^{(i)}\}_{i=1}^{N}$。这样，对于一个特定的抗体可以计算得到一个相应于目标函数（抗原）的亲和度（抗体种群 $Ab_{k,\{N\}}$ 中的每个抗体 $Ab_k^{(i)}$ 表示输入空间的一个元素）。

步骤 3：克隆选择。

① 确定抗体种群 $Ab_{k,\{N\}}$ 中所有 N 个抗体关于抗原（目标函数）的亲和度 aff。

② 从 $Ab_{k,\{N\}}$ 中选择 N_1 个亲和度最高的抗体构成一个新的关于抗原的高亲和度抗体集合 $Ab_{k,\{N_1\}}=\{Ab_k^{(j)}\}_{j=1}^{N_1}$。

③ 对步骤 3②所选择的抗体种群 $Ab_{k,\{N_1\}}$ 中的抗体分别进行复制。

由步骤 3②选择的各抗体的克隆个数与其相应于抗原的亲和度成正比，即对于 N_1 个所选抗体，与抗原亲和度越高的抗体其克隆的个数越多，从而产生一个克隆抗体种群 Cl。

步骤 4：抗体库更新。

① 克隆变异。克隆抗体在分裂的过程中会发生随机点变异，且其中的每个抗体发生变异的概率与其抗原亲和度成反比，即克隆抗体中与抗原亲和度越高的抗体变异率越低，产生一个变异的克隆抗体种群 Cl^*。

② 亲和度计算。确定变异克隆抗体种群 Cl^* 中每个抗体关于抗原的亲和度 aff^*。

③ 克隆选择。从变异克隆种群 Cl^* 中重新选择 N 个亲和度最高的抗体 $Ab_{k,\{N\}}^*$。$Ab_{k,\{N\}}^*$ 作为候选抗体进入记忆抗体种群，这里 $Ab_{k,\{N\}}^*$ 为克隆扩增抗体种群。

④ 受体编辑。模拟生物免疫系统的受体编辑过程随机产生 N_2 个全新抗体构成新抗体种群 $Ab_{k,\{N_2\}}$，并用 $Ab_{k,\{N_2\}}$ 中的 N_2 个新抗体替换 $Ab_{k,\{N\}}^*$ 中 N_2 个亲和度较低的抗体构成下一代抗体种群 $Ab_{k+1,\{N\}}$。

步骤 5：种群评估。

① 计算种群 $Ab_{k+1,\{N\}}$ 的平均亲和度；

② 判断抗体库是否满足成熟条件，若已成熟则转步骤 7；否则转步骤 6。

步骤 6：置 k=k+1，重复步骤 3～步骤 5。

步骤 7：当满足迭代终止条件时结束循环。

循环终止的条件应满足使抗体种群的平均亲和度达到一个稳定值。这时，从循环迭代最终产生的抗体种群中选择出亲和度最高的抗体作为最优解。

10.2.2 人工免疫粒子滤波算法

通过前文的分析可知，用于表示外来侵犯的抗原和免疫系统产生的抗体分别与待求解问题的目标函数和候选解相对应。基于这一原理，若将被估计的状态与抗原相对应，用于状态近似的粒子与抗体相对应，即可实现免疫算法与粒子滤波的结合，得到人工免疫粒子滤波算法，具体思路如下：

根据重要密度函数随机抽取 N 个样本作为初始抗体，这 N 个抗体实际上也是状态估计时所要用到的粒子；对这 N 个粒子进行更新，分别计算 N 个抗体的亲和度和抗体与抗体之间的排斥力，按照亲和度的大小确定每个抗体应克隆的抗体数目。N 个初始抗体和克隆所得的所有抗体组成新的抗体群，通过其亲和度的计算，可舍弃相似的抗体，并从中选出最好的 N 个抗体用于对下一时刻的状态估计。这种做法必然会使计算量有所增加，但在条件允许的情况下，可使样本集保持一定的多样性，减轻退化现象的影响，将优选后的粒子用于后验概率密度的近似，可有效提高状态估计性能。

若以 1 和粒子集 $\{\boldsymbol{x}_k^{(i)}, w_k^{(i)}\}$ 分别作为抗原和抗体，可得到如下几个重要定义。

定义 1　k 时刻第 i 个抗体的亲和度 $aff_k(i)$ 定义为

$$aff_k(i)=1-w_k^{(i)} \tag{10-3}$$

$aff_k(i)$ 越小，表明抗体和抗原匹配得越好。

定义 2　k 时刻第 i 个抗体与第 j 个抗体的排斥力 $off_k(i,j)$ 定义为

$$off_k(i,j)=\left\|\boldsymbol{x}_k^{(i)}-\boldsymbol{x}_k^{(j)}\right\| \tag{10-4}$$

$off_k(i,j)$ 越小，表明第 i 个抗体与第 j 个抗体越相似。

定义 3　k 时刻第 i 个抗体的克隆数 $\text{knum}_k(i)$ 定义为

$$\text{knum}_k(i)=\text{round}\left(N\times\cos[\frac{\pi}{2}\times aff_k(i)]\right) \tag{10-5}$$

式中：round(·) 表示向最近的整数取整。选用余弦函数的目的是为使亲和度小的粒子克隆

数目较大，并且分布在0~N之间。克隆的原则是促进亲和度小的抗体，而抑制亲和度大的抗体，以使系统能快速收敛于全局最优解。

定义 4　k时刻第i个抗体的变异规则定义为

$$x_k^{(i)} = x_k^{(i)} + aff_k(i)\cdot\zeta \tag{10-6}$$

式中：$\zeta\sim\mathcal{N}(x;0,\Sigma)$为服从高斯分布的随机量。变异的原则是对亲和度小的抗体赋予较小的变异量，而对亲和度大的抗体则赋予较大的变异量。

由此可得免疫粒子滤波（AIPF）算法流程如下。

Alg（10.3）：Artificial Immune Particle Filtering

步骤 1：初始化。在$k=0$时刻，根据重要密度函数抽样出N个粒子，即$\{x_k^{(i)},1/N\}_{i=1}^{N}$。

步骤 2：更新。估计k时刻的粒子$x_k^{(i)}$及其归一化权值，并将这N个粒子作为初始抗体群存入记忆库。

步骤 3：计算亲和度。按照式（10-3）和式（10-4）计算每个抗体的亲和度$aff_k(i)$及抗体与抗体之间的排斥力$off_k(i,j)$。

步骤 4：克隆抗体。按照每个抗体的亲和度对抗体进行克隆，克隆数目$\mathrm{knum}_k(i)$由式（10-5）确定。

步骤 5：变异。按照式（10-6）对所有抗体进行变异。

步骤 6：选优。计算变异后所有抗体的$aff_k(i)$和$off_k(i,j)$，抛弃$off_k(i,j)<Th$中的一个抗体，并按$aff_k(i)$进行排序，选出前面的N个抗体更新记忆库。返回步骤 3，直到满足终止判据为止。

步骤 7：估计。用记忆库中的N个抗体作为新一代的粒子，计算$k+1$时刻的状态估计，并进行重采样。

步骤 8：令$k=k+1$，返回步骤 2。

由算法的实现过程看，在每一时刻k，可能涉及抗体群的多次迭代和更新，从而使粒子得以优选，提高了对系统状态的估计能力。

10.3　人工鱼群粒子滤波

10.3.1　人工鱼群算法原理

人工鱼群算法（AFSA）[63-65]是一种通过模拟鱼群在自然环境中的生态行为得到的随机搜索优化算法，这种算法不需要问题的严格机理模型，从而使其具有广泛的应用范围。人工鱼是真实鱼个体的一个虚拟实体，这个实体封装了人工鱼自身数据信息和一系列的行为。人工鱼所在的环境主要是问题的解空间和其他人工鱼的状态，它在下一时刻的行为取决于目前自身的状态和目前环境的状态（包括问题当前解的优劣和其他同伴的状态），并通过自身的活动来影响环境，进而影响其他同伴的活动。

AFSA 是一种集群智能算法。一个集群通常定义为一群自治体的集合，它们通过相互直接或者间接的通信及全体的活动来解决一些分布式难题。这里，自治体是指在一个环境中具备自身活动能力的实体，其自身力求简单，通常不必具有高级智能，但它们的

集群活动所表现出来的则是一种高级智能才能达到的活动，称这种活动为集群智能。

假设虚拟人工鱼实体的当前状态为 $\boldsymbol{x}$，Visual 为其视野范围，状态 $\boldsymbol{x}_v$ 为其在某时刻视点所在的位置。如果该位置的状态优于当前状态，则考虑向该位置方向前进一步，即到达状态 $\boldsymbol{x}_{\text{next}}$；如果状态 $\boldsymbol{x}_v$ 不比当前状态更优，则继续巡视视野内的其他位置。巡视的次数越多，则对视野内的状态了解更全面，从而对周围的环境有一个全方位立体的认知，这有助于做出相应的判断和决策。当然，对于状态多或无限状态的环境也不必全部遍历，允许一定的不确定性对于摆脱局部最优、寻找全局最优是有帮助的。假设状态 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)$，$\boldsymbol{x}_v=(x_{v,1},x_{v,2},\cdots,x_{v,n})$，该过程可以表示为

$$\boldsymbol{x}_{v,i}=\boldsymbol{x}_i+\text{Visual}\cdot u$$

$$\boldsymbol{x}_{\text{next}}=\frac{\boldsymbol{x}_v-\boldsymbol{x}}{\|\boldsymbol{x}_v-\boldsymbol{x}\|}\cdot\text{StepSize}\cdot u$$

式中：$u\sim\mathcal{U}[0,1]$ 为 0～1 的随机数；StepSize 为移动步长。由于环境中同伴的数目是有限的，因此，在视野中感知同伴的状态，并相应地调整自身状态的方法和上式类似。

在 AFSA 中，一个重要的概念是鱼群的“行为模式”，它表征了鱼类各种活动的表现形式，通常包括以下几种。

（1）觅食行为（Prey behaviour）。这是生物的一种最基本的行为，也就是趋向食物的一种活动。一般情况下，鱼在水中随机地自由游动，当发现食物时，则会向食物逐渐增多的方向快速游去，可以认为它是通过视觉或味觉来感知水中的食物量或浓度来选择趋向的。因此，以上所述的视觉概念可以应用于该行为。

（2）聚群行为（Swarm behaviour）。这是鱼类较常见的一种现象，大量或少量的鱼都能聚集成群，这是它们在进化过程中形成的一种生存方式。鱼在游动过程中为了保证自身的生存和躲避危害会自然地聚集成群，鱼聚群时所遵守的规则有三条：①分隔规则。尽量避免与临近伙伴过于拥挤。②对准规则。尽量与临近伙伴的平均方向一致。③内聚规则。尽量朝临近伙伴的中心移动。

（3）追尾行为（Follow behaviour）。当鱼群中的一条或几条鱼发现食物时，其临近的伙伴会尾随其快速到达食物点。

（4）随机行为。鱼在水中悠闲地自由游动，基本上是随机的，其目的是为了在更大范围内寻觅食物或伙伴。

每条人工鱼通过对环境的感知，在每次移动中经过尝试后，执行其中的一种行为。人工鱼群算法就是利用这几种典型行为从构造单条鱼底层行为做起，通过鱼群中每一个体局部寻优达到全局最优化的目的。算法的进行就是人工鱼个体的自活动过程，整个过程主要包括觅食、聚群以及追尾等几种行为，最优解将在该过程中突现出来。其中觅食行为是人工鱼根据当前自身的适应值随机游动的行为，是一种个体极值寻优过程，属于自学习的过程；而聚群和追尾行为则是人工鱼与周围环境交互过程。这两种过程是在保证不与伙伴过于拥挤，且与临近伙伴的平均移动方向一致的情况下向群体极值（中心）移动。

如何利用简便、有效的方式来构造并实现觅食、聚群、尾追等行为是 AFSA 实施的关键，下面分别进行介绍。

1. 觅食行为

假设人工鱼个体的状态可用矢量表示为 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)$，其中，$\boldsymbol{x}_i(i=1,\cdots,n)$ 为欲寻优的变量；人工鱼当前所在位置的食物浓度表示为 $y=f(\boldsymbol{x})$，其中，y 为目标函数值；人工鱼个体之间的距离表示为 $d_{ij}=\|\boldsymbol{x}_i-\boldsymbol{x}_j\|$；$v$ 表示人工鱼的感知距离。

设人工鱼当前状态为 $\boldsymbol{x}_i$，在其感知范围内（$d_{ij}<v$）随机选择一个状态 $\boldsymbol{x}_j$，若 $y_i<y_j$，则向该方向前进一步；若不满足，则随机移动一步，即

$$\text{if } y_i<y_j,\ \boldsymbol{x}_{i,\text{next}}=\boldsymbol{x}_i+r\cdot s\cdot\frac{\boldsymbol{x}_j-\boldsymbol{x}_i}{\|\boldsymbol{x}_j-\boldsymbol{x}_i\|}$$

$$\text{else } \boldsymbol{x}_{i,\text{next}}=\boldsymbol{x}_i+\boldsymbol{r}\cdot s$$

式中：s 表示人工鱼移动的最大步长；$\boldsymbol{r}\sim\mathcal{U}[\mathbf{0},\mathbf{1}]$ 为维数与 $\boldsymbol{x}_i$ 相同的随机矢量；$r\sim\mathcal{U}[0,1]$ 为介于 0～1 的一个随机数。

2. 聚群行为

人工鱼在游动过程中自然聚集成群，当在其感知范围内（$d_{i,j}<v$）探索其他集群伙伴数目 N_f 和中心位置 $\boldsymbol{x}_c=\dfrac{1}{N_f}\sum_{j=1}^{N_f}\boldsymbol{x}_j$ 时，若 $y_c/N_f>\delta y_i$，表明伙伴中心有很多食物并且不太拥挤，则向伙伴中心移动，即

$$\frac{y_c}{N_f}>\delta y_i,\ \boldsymbol{x}_{i,\text{next}}=\boldsymbol{x}_i+r\cdot s\cdot\frac{\boldsymbol{x}_c-\boldsymbol{x}_i}{\|\boldsymbol{x}_c-\boldsymbol{x}_i\|}$$

否则，执行觅食行为。式中，δ 为拥挤度因子。

3. 追尾行为

当一条鱼发现食物后，其他人工鱼会尾随它向食物游去。搜索当前领域中食物浓度最大的人工鱼所在位置 $y_{\max}$，当 $\dfrac{y_{\max}}{N_f}>\delta y_i$ 时，则向食物浓度最大的人工鱼处前进一步，即

$$\boldsymbol{x}_{i,\text{next}}=\boldsymbol{x}_i+r\cdot s\cdot\frac{\boldsymbol{x}_{\max}-\boldsymbol{x}_i}{\|\boldsymbol{x}_{\max}-\boldsymbol{x}_i\|}$$

否则，执行觅食行为。

4. 行为选择

根据所要解决的问题性质，对人工鱼当前所处的环境进行评价，从上述各行为中选择一种合适的行为。常用的方法有两种。

（1）先进行追尾行为，如果没有进步则进行聚群行为，如果依然没有进步则进行觅食行为。也就是选择较优的行为前进，即任选一种行为，只要能向更优的方向前进即可。

（2）试探执行各种行为，选择各行为中使得向最优方向前进最快的行为，即模拟聚群、追尾等行为，然后选择能导致该行动后使状态更优的动作，缺省的行为方式是觅食行为。也就是选择各行为中使得人工鱼的下一个状态最优的行为，如果没有能使人工鱼的下一状态优于当前状态的行为，则采取随机行为。对于此种方法，同样的迭代步数寻优效果会好一些，但计算量也会加大。

人工鱼群算法通过这些行为的选择形成了一种高效的寻优策略，最终，人工鱼将集结在目标函数的极值区域。AFSA 的基本流程如图 10.3 所示。

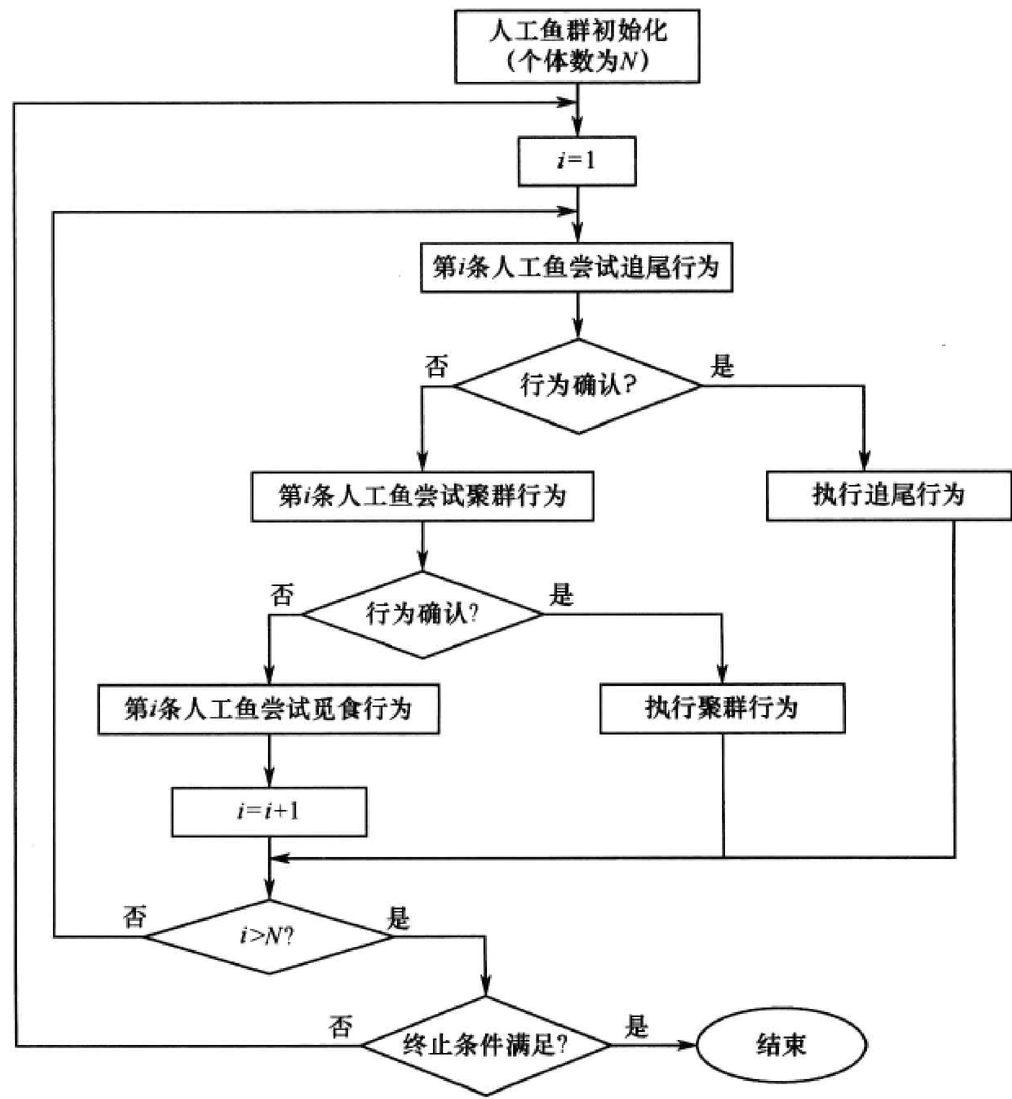


图 10.3 人工鱼群算法流程图

10.3.2 人工鱼群粒子滤波算法

若将人工鱼的个体与粒子对应，将系统的观测方程与目标函数对应，在传统的粒子滤波算法中融入 AFSA 的聚群行为和觅食行为所对应的粒子状态更新公式，即可得到人工鱼群粒子滤波（AFSPF）算法，具体步骤可概括如下：

Alg（10.4）：Artificial Fish Swarm Particle Filtering

步骤 1：初始化

在 $k=0$ 时刻，从重要性函数中采样得到 N 个粒子，即 $\{\boldsymbol{x}_0^{(i)},1/N\}_{i=1}^{N}$。

步骤 2：粒子时间更新

通过采样函数 $q(\boldsymbol{x}_k^{(i)}\mid\boldsymbol{x}_{k-1}^{(i)},\boldsymbol{z}_k)$ 产生粒子，若采样函数为先验分布，即

$$q(\boldsymbol{x}_k^{(i)}\mid\boldsymbol{x}_{k-1}^{(i)},\boldsymbol{z}_k)=p(\boldsymbol{x}_k^{(i)}\mid\boldsymbol{x}_{k-1}^{(i)})$$

则 $\boldsymbol{x}_k^{(i)}=(\hat{x}_{1,k|k-1},\hat{x}_{2,k|k-1},\cdots,\hat{x}_{n_x,k|k-1})\sim p(\boldsymbol{x}_k^{(i)}\mid\boldsymbol{x}_{k-1}^{(i)})$。

步骤 3：重要权值计算

$$w_k^{(i)} = w_{k-1}^{(i)} p(z_k \mid x_k^{(i)})$$

令目标函数 $y = p_v(z_k - \hat{z}_{k|k-1}^{(i)})$，若 $\boldsymbol{v}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_k)$，则

$$y = \frac{1}{\sqrt{(2\pi)^{n_z} \det(\boldsymbol{R}_k)}} \exp\left[-\frac{1}{2}(z_k - \hat{z}_{k|k-1}^{(i)})^T \boldsymbol{R}_k^{-1}(z_k - \hat{z}_{k|k-1}^{(i)})\right]$$

（1）觅食行为。当 $y_i < y_j$ 时，通过与目标函数进行比较，不断更新粒子的状态使其向真实值靠近。

$$\hat{\boldsymbol{x}}_{k|k-1}^{(i_m)} = \hat{\boldsymbol{x}}_{k|k-1}^{(i_{m-1})} + r \cdot s \cdot \frac{\hat{\boldsymbol{x}}_{k|k-1}^{(j)} - \hat{\boldsymbol{x}}_{k|k-1}^{(i_{m-1})}}{\left\| \hat{\boldsymbol{x}}_{k|k-1}^{(j)} - \hat{\boldsymbol{x}}_{k|k-1}^{(i_{m-1})} \right\|}$$

式中：$\hat{\boldsymbol{x}}_{k|k-1}^{(i_m)}$ 为对 $\hat{\boldsymbol{x}}_{k|k-1}^{(i)}$ 第 m 次（$m=1,2,\cdots,M$）迭代后的结果。否则，$\hat{\boldsymbol{x}}_{k|k-1}^{(i_m)} = \hat{\boldsymbol{x}}_{k|k-1}^{(i_{m-1})} + \boldsymbol{r} \cdot s$。

（2）聚群行为。设 $\hat{z}_{k|k-1}^{(i)}$ 在范围 v 内的其他预测量测值集合为 A_v，$|A_v|$ 为集合 A_v 的势，则其中心为 $\hat{z}_{k|k-1}^{(ic)} = \frac{1}{|A_v|}\sum_{j \neq i}^{|A_v|} \hat{z}_{k|k-1}^{(j)}$。若 $\frac{y_{ic}}{|A_v|} > \delta \cdot y_i$，则 $\hat{\boldsymbol{x}}_{k|k-1}^{(i_m)} = \hat{\boldsymbol{x}}_{k|k-1}^{(i_{m-1})} + r \cdot s \cdot \frac{\hat{\boldsymbol{x}}_{k|k-1}^{(ic)} - \hat{\boldsymbol{x}}_{k|k-1}^{(i_{m-1})}}{\left\| \hat{\boldsymbol{x}}_{k|k-1}^{(ic)} - \hat{\boldsymbol{x}}_{k|k-1}^{(i_{m-1})} \right\|}$，否则转觅食行为。

算法的终止可以通过限制最大迭代次数 M 或者以相邻两次迭代中目标函数的偏差小于某一预设的门限为条件。

步骤 4：权值更新

根据 $w_k^{(i)} = w_{k-1}^{(i)} p(z_k \mid x_k^{(i)})$ 更新权值，并通过 $\tilde{w}_k^{(i)} = w_k^{(i)} / \sum_{i=1}^{N_1} w_k^{(i)}$ 对取出的 N_1 个粒子的权值进行归一化。

步骤 5：估计输出

状态估计：$\hat{\boldsymbol{x}}_k = \sum_{i=1}^{N_1} \tilde{w}_k^{(i)} \boldsymbol{x}_k^{(i)}$

方差估计：$\boldsymbol{P}_k = \sum_{i=1}^{N_1} \tilde{w}_k^{(i)} (\boldsymbol{x}_k^{(i)} - \hat{\boldsymbol{x}}_k)(\boldsymbol{x}_k^{(i)} - \hat{\boldsymbol{x}}_k)^{\mathrm{T}}$

步骤 6：判断是否结束

若是则退出本算法，若否则返回步骤 2。

10.4 本 章 小 结

粒子滤波算法的性能取决于对状态后验分布的近似精度，而近似精度又跟粒子的数目、质量（有效性、多样性）等因素密切相关。因此，为了提高粒子滤波算法的估计性能，一条重要的途径就是提高粒子的产生质量，避免粒子退化和贫化。在第 8 章中，主要阐述通过对不同重要性采样函数的近似来达到提高粒子产生质量的目的。本章则从另一个角度，即通过将人工智能优化选择与粒子滤波相结合的手段，重点介绍了遗传粒子滤波、人工免疫粒子滤波和人工鱼群粒子滤波的算法原理和实现方法。这类滤波算法的

相似之处是利用了人工智能的优化策略，在粒子产生和更新的过程中增加了优选环节，保持了粒子的多样性，同时使其往高似然区域移动和聚集。

应该指出的是，作为提高粒子滤波性能的一条重要途径，基于最优化理论的粒子优选方法越来越受到重视。理论上，除了前文涉及的几种重要方法外，人工智能与最优化理论中的其他一些方法都具备与粒子滤波结合的潜力，且各种方法的组合优化可望进一步提高粒子滤波的整体估计性能，关于这一点，可参阅文献[66]～文献[68]。

参 考 文 献

[1] Gordon N J，Salmon D J, Smith A F M. Novel approach to nonlinear/non-Gaussian Bayesian state estimation[J]. IEE Proceedings-Radar & Signal Processing, 1993, 140（2）: 107–113.

[2] Fearnhead P. Sequential Monte Carlo methods in filter theory[D]. University of Oxford, 1998.

[3] Liu J S, Chen R. Sequential Monte Carlo methods for dynamic systems[J]. Journal of the American Statistical Association, 1998, 93（443）: 1032–1043.

[4] Doucet A, De Freitas N, Gordon N. Sequential Monte-Carlo Methods in Practice[M]. Springer-Verlag, 2001.

[5] Andrieu C, Doucet A. Particle filtering for partially observed Gaussian state space models[J]. Journal of the Royal Statistical Society-Series B, 2002, 64（4）: 827–836.

[6] Djuric P M, Kotecha J H, Zhang J, et al. Particle filtering[J]. IEEE Signal Processing Magazine, 2003, 20（5）: 19–38.

[7] Doucet A, Gordon N J, Krishnamurthy V. Particle filters for state estimation of jump Markov linear systems[J]. IEEE Transaction on Signal Processing, 2001, 49（3）: 613–624.

[8] 胡士强, 敬忠良. 粒子滤波算法综述[J]. 控制与决策, 2005, 20（4）: 361–365.

[9] Crisan D, Doucet A. A Survey of convergence results on particle filtering methods for practitioners[J]. IEEE Transactions on Speech and Audio Processing, 2002, 10（3）: 173–185.

[10] Doucet A, Godsill S, Andrieu C. On sequential Monte Carlo sampling methods for Bayesian filtering[J]. Statistics and Computing, 2000, 10（3）: 197–208.

[11] Douc R, Cappe O, Moulines E. Comparison of resampling schemes for particle filtering[C]. Proceedings of the International Symposium on Image and Signal Processing and Analysis, 2005: 64–69.

[12] Doucet A. On sequential Monte Carlo methods for Bayesian filtering[R]. Cambridge University Engineering Department, 1998.

[13] Kong A, Liu J S, Wong W H. Sequential imputations and Bayesian missing data problems[J]. Journal of the American Statistical Association, 1994, 89（425）: 278–288.

[14] Arulampalam S, Maskell S, Gordan N, et al. A tutorial on particle filter for on-line nonlinear/non-Gaussian Bayesian tracking[J]. IEEE Transactions on Signal Processing, 2002, 50（2）: 174–188.

[15] Carpenter J, Clifford P, Fearnhead P. Improved particle filter for nonlinear problems[J]. IEE Proceedings-Radar, Sonar & Navigation, 1999, 146（1）: 2–7.

[16] Higuchi T. Monte Carlo filtering using genetic algorithm operators[J]. Journal of Statistical Computation and Simulation, 1997, 59（1）: 1–23.

[17] Kitagawa G. Monte Carlo filter and smoother for non-Gaussian nonlinear state-space[J]. Journal of Computational and Graphical Statistics, 1996, 5（1）: 1–25.

[18] Gilks W R, Berzuini C. Following a moving target-Monte Carlo inference for dynamic Bayesian models[J]. Journal of the Royal Statistical Society-Series B, 2001, 63（1）: 127–146.

[19] Fearnhead P. MCMC, sufficient statistics and particle filter[J]. Journal of Computuational and Graphical Statistics, 2002, 11(4): 848–862.

[20] Carlin B P, Polson N G, Stoffer D S. A Monte Carlo approach to nonnormal and nonlinear state-space modeling[J]. Journal of the American Statistical Association, 1992, 87（418）: 493–500.

[21] Berzuini C, Best N G, Gilks W, et al. Dynamic conditional independence models and Markov chain Monte Carlo methods[J]. Journal of the American Statistical Association, 1997, 92（5）: 1403–1412.

[22] 袁泽剑, 郑南宁, 贾新春. 高斯—厄米特粒子滤波器[J]. 电子学报, 2003, 31（7）: 970–973.

[23] Pitt M, Shephard N. Filtering via simulation: auxiliary particle[J]. Journal of the American StatisticalAssociation, 1999, 94（446）: 590–599.

[24] Van De Merwe R, De Freitas N, Doucet A, et al. The unscented particle filter[R]. Cambridge University Engineering Department, 2000.

[25] Mahler R P S. Multitarget Bayes filtering via first order multitarget moments[J]. IEEE Transactions on Aerospace and Electronic Systems, 2003, 39（4）: 1152–1178.

[26] Punithakumar K, Kirubarajan T, Sinha A. A sequential Monte Carlo probability hypothesis density algorithm for multitarget track-before-detect[C]. Proceedings of SPIE, 2005, 5913: 59131S-1–59131S-8.

[27] B-N Vo, Singh S, Doucet A. Sequential Monte Carlo methods for multitarget filtering with random finite sets[J]. IEEE Transactions on Aerospace and Electronic Systems, 2005, 41（4）: 1224–1245.

[28] Dominic S, Batuong V, B-N Vo. A consistent metric for performance evaluation of multi-object filters[J]. IEEE Transactions on Signal Processing, 2008, 56（8）: 3447–3457.

[29] 赵欣, 姬红兵, 杨柏胜. 基于随机有限集的 RBPF 多目标关联跟踪算法[J]. 电子学报, 2011, 39（3）: 505–510.

[30] Robert C P, Casella G. Monte Carlo Statistical Methods[M]. New York: Springer-Verlag, 1999.

[31] 占荣辉, 辛勤, 万建伟. 基于最优采样函数的粒子滤波算法与贝叶斯估计[J]. 信号处理, 2008, 24（2）: 259–263.

[32] 占荣辉. 基于空频域信息的单站被动目标跟踪算法研究[D]. 长沙：国防科学技术大学博士学位论文, 2007.

[33] Xiao-long D, Jian-ying X, Hong-wei N. Interacting multiple model algorithm with the unscented particle filter （UPF）[J]. Chinese Journal of Aeronautics, 2005, 18（4）: 366–371.

[34] Rui Y, Chen Y. Better proposal distributions: Object tracking using unscented particle filter[C]. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2001: 786–793.

[35] 袁健, 张文霞, 隋树林. 一种目标轮廓跟踪的 UPF 方法[J]. 青岛科技大学学报（自然科学版）, 2006, 27（4）: 355–358.

[36] Ronghui Z, Qin X, Jianwei W. A modified unscented particle filter for nonlinear Bayesian tracking[J]. Journal of Systems Engineering and Electronics, 2008, 19（1）: 7–14.

[37] Alspach D L, Sorenson H W. Nonlinear Bayesian estimation using Gaussian sum approximations[J]. IEEE Transactions on Automatic Control, 1972, 17（4）: 439–448.

[38] Caputi M J, Moose R L. A modified Gaussian sum approach to estimation of non-Gaussian signals[J]. IEEE Transactions on Aerospace and Electronic Systems, 1993, 29（2）: 446–451.

[39] Tam W I, Plataniotis K N, Hatzinakos D. An adaptive Gaussian sum algorithm for target tracking[J]. Signal Processing, 1999, 77（1）: 85–104.

[40] Kotecha J H, Djuric P M. Gaussian particle filtering[J]. IEEE Transactions on Signal Processing, 2003, 51（10）: 2592–2601.

[41] Jinxia Y, Yongli T, Xiancha C, et al. Choice mechanism of proposal distribution in particle filter[C]. Proceedings of the 8th World Congress on Intelligent Control and Automation, 2010: 1051–1056.

[42] Schon T B, Gustafsson F, Nordlund P. Marginalized particle filters for mixed linear/nonlinear state-space models[J]. IEEE Transactions on Signal Processing, 2005, 52（7）: 2279–2289.

[43] Eidehall A, Schon T B, Gustafsson F. The marginalized particle filter for automotive tracking applications[C]. Proceedings of IEEE Intelligent Vehicle Symposium, 2005: 369–374.

[44] Li P, Goodall R M, Kadirkamanathan V. Estimation of parameters in a linear state space model using a Rao-Blackwellised particle filter[J]. IEE Proceedings-Control Theory & Application, 2004, 151（6）: 727–738.

[45] Xu X, Li B. Adaptive Rao-Blackwellized particle filter and its evaluation for tracking in surveillance[J]. IEEE Transactions on Image Processing, 2007, 16（3）: 838–849.

[46] Vihola M. Rao-blackwellised particle filtering in random set multitarget tracking[J]. IEEE Transactions on Aerospace and Electronic Systems, 2007, 43（2）: 689–705.

[47] Schon T B, Karlsson R, Gustafsson F. Complexity analysis of the marginalized particle filter[J]. IEEE Transactions on Signal Processing, 2005, 53（11）: 4408–4411.

[48] Ronghui Z, Ling W, Jianwei W, et al. Passive target tracking using marginalized particle filter[J]. Journal of Systems

Engineering and Electronics, 2007, 18（3）: 503–508.

[49] Rutten M G, Gordon N J, Maskell S. Particle-based track-before-detect in Rayleigh noise[C]. Proceedings of SPIE, 2004, 5428: 509–519.

[50] Rutten M G, Ristic B, Gordon N J. A comparison of particle filters for recursive track-before-detect[C]. Proceeding of the 8th International Conference on Information Fusion, 2005: 169–175.

[51] 龚亚信, 杨宏文, 胡卫东, 等. 基于粒子滤波的弱目标检测前跟踪算法[J]. 系统工程与电子技术, 2007, 29（12）: 2143–2148.

[52] Higuchi T. Monte Carlo filtering using genetic algorithm operators[J]. Journal of Statistical Computation and Simulation, 1997, 59: 1–23.

[53] 叶龙, 王京玲, 张勤. 遗传重采样粒子滤波器[J]. 自动化学报, 2007, 33（8）: 885–887.

[54] Hua H, Yong-Sheng D, Kuang-Rong H, et al. An evolutionary particle filter with the immune genetic algorithm for intelligent video target tracking[J]. Computers and Mathematics with Applications, 2011, 62: 2685–2695.

[55] Park S, Hwang J P, Kim E, et al. A new evolutionary particle filter for the prevention of sample impoverishment[J], IEEE Transactions on Evolutionary Computation, 2009, 13（4）: 801~809.

[56] 莫以为, 萧德云. 进化粒子滤波算法及其应用[J]. 控制理论与应用, 2005, 22（2）: 269–270.

[57] Park S, Hwang J, Rou K, et al. A new particle filter inspired by biological evolution: genetic filter[J]. International Journal of Applied Science Engineering and Technology, 2007, 4 （1）: 459–463.

[58] 张琪, 王鑫, 胡昌华, 等. 人工免疫粒子滤波算法的研究[J]. 控制与决策, 2008, 23（3）: 293–296.

[59] Hua H, Yongsheng D, Kuangrong H. A new immune particle filter algorithm for tracking a moving target[C]. Proceedings of 6th International Conference on Natural Computation, 2010: 3248~3252.

[60] Akhtar S, Ahmad A R, Abdel-Rahman E M, et al. A PSO accelerated immune particle filter for dynamic state estimation[C]. Proceedings of Canadian Conference on Computer and Robot Vision, 2011: 72–79.

[61] 陆陶荣, 朱林户, 李德芳, 等. 一种集群智能粒子滤波算法[J]. 西安电子科技大学学报（自然科学版）, 2008, 35（3）: 536–541.

[62] Yu-min T, Li C. Unscented particle filter algorithm based on artificial fish swarm algorithm[C]. Proceedings of 8th International Conference on Natural Computation, 2012: 1123–1126.

[63] 李晓磊, 邵之江, 钱积新. 一种基于动物自治体的寻优模式: 鱼群算法[J]. 系统工程理论与实践, 2002, 22（11）: 32–38.

[64] 李晓磊. 一种新型智能优化算法——人工鱼群算法[D]. 杭州: 浙江大学博士学位论文, 2003.

[65] 张梅凤. 人工鱼群智能优化算法的改进及应用研究[D]. 大连: 大连理工大学博士学位论文, 2008.

[66] Yi-Tung K, Erwie Z. A hybrid genetic algorithm and particle swarm optimization for multimodal functions[J]. Applied Soft Computing, 2008, 8: 849–857.

[67] 基于多智能体协同进化的粒子滤波目标跟踪算法[J].模式识别与人工智能, 2011, 24（1）: 57–62.

[68] 周殊, 潘炜, 罗斌, 等. 一种基于粒子群优化方法的改进量子遗传算法及应用[J]. 电子学报, 2006, 34（5）: 897–901.

第四部分　非线性滤波的扩展与应用

在贝叶斯滤波的统一框架下，本书第二部分和第三部分从不同的近似角度，阐述了一般条件下的非线性/非高斯滤波问题，主要基于两点基本的假设：①系统模型只有单一模式，即虽然系统状态是动态变化的，但在所讨论的时间范围内其演化规则是唯一的；②基于各种方法的系统状态估计是在单传感器观测条件下得到的，即同一时刻尽管因为杂波或噪声的影响存在多个量测（包括虚警），但这些观测信息源于单一传感器。

但是，现实世界中面临的问题往往比以上假设条件更为复杂，如对于目标跟踪而言，目标的机动会导致其运动模式（或运动状态演化规则）的变化，而无论是对于民用还是军事目标，出于特定的目的需要，执行相应的机动动作是非常普遍的；另外，为了提高对目标的跟踪能力，采用单/多平台多传感器协同处理也是常用的手段。在这种条件下，前述的滤波理论和方法又面临新的挑战。

本部分主要以第二部分和第三部分的理论和方法为基础，进一步分析两种扩展条件，即多传感器观测和混合系统模型条件下的非线性滤波及其应用问题。第 11 章主要针对集中式融合和分布式融合两种结构，介绍多传感器融合估计方法；第 12 章则主要针对混合系统的估计问题，重点介绍多模滤波算法。每章最后给出了基于解析高斯滤波和近似采样滤波的目标跟踪应用实例。

第 11 章　多传感器融合滤波

目前，在军事和民用领域中，目标的监视、检测、跟踪以及识别处理等很多都是基于多传感器的量测，利用信息融合（Information Fusion）技术来完成的。所谓信息融合[1-5]就是将来自多个传感器或多源的信息进行综合处理，用以得出更为准确、可靠的结论。

与单传感器系统相比较，多传感器信息融合系统尽管增加了系统的复杂性，但具有单传感器系统所不具备的诸多优势，如可有效扩展信息的时间、空间和频率（谱段）覆盖范围，增加量测空间维数，降低信息的模糊度，提高处理结果可信度等；此外，特定的融合结构还可增强系统的生存能力，改善系统的可靠性。

融合滤波的一个重要目的就是综合利用多个传感器的量测信息，来提高系统状态估计精度。对于一些特殊的应用，如只测角跟踪中，在单传感器配置条件下，所能够获取的目标信息将会非常有限，加上受可观测性条件的限制，严重限制了被动系统对目标的探测与跟踪能力。利用多传感器信息进行融合处理可望获得较大的能力提升，不仅可以降低对观测器的机动性要求，而且能够得到更加精确的目标航迹估计。

本章首先给出了不同融合结构中的贝叶斯滤波统一框架，进而讨论线性、高斯条件下的最优融合算法，并将其推广到非线性系统中，最后通过具体的实例对融合算法进行了仿真验证。

11.1　融合估计系统结构

采用多传感器观测进行融合估计处理，有着多种实现途径，按照融合结构的不同，可以分为集中式融合（Centralized Fusion）、分布式（Distributed Fusion）融合以及混合式融合（Hybrid Fusion）融合结构。

集中式融合结构如图 11.1 所示，在这种处理结构中，融合中心可以利用所有传感器的原始量测数据（故也称为量测融合（Measurement Fusion）），没有任何信息的损失，因而理论上这种融合结果是最优的。但是这种结构同样存在一些自身的问题，主要体现在以下几个方面：

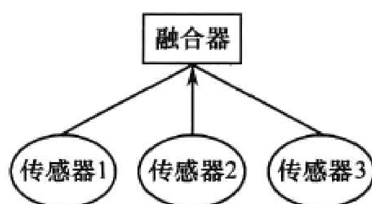


图 11.1　集中式结构

（1）原始量测信息需直接传送至融合中心，对通信带宽和实时传输能力要求较高。

（2）融合中心要具备强大的计算能力，以满足大数据量处理要求。

（3）在异质传感器条件下，传感器原始观测数据形式不同，难以直接进行融合。

（4）传感器原始观测数据的配准难度较大，无论是时间配准还是空间配准。

除此之外，由于融合处理过程过分依赖于融合中心，使得集中式融合系统的生存能力也下降。

分布式融合结构如图 11.2 所示，在这种结构中，每个局部传感器（节点）都有自己的处理器，经自身处理后将中间结果送到融合中心。由于各传感器都有自己的局部处理器，能够得到各自的滤波估计结果，因此这种结构通常也称为航迹融合（Track Fusion），融合中心得到全局的估计结果。与集中式融合相比，分布式融合得到的结果通常是次优的；但这种融合方式因具有对通信要求低、系统生命力强、工程实现相对容易等特点而广受关注。

典型的混合式融合结构如图 11.3 所示，顾名思义，这种融合结构是集中式和分布式融合的混合体，融合中心得到的部分为原始量测数据，部分为经由某些局部节点处理过的中间数据。由于这种结构兼具集中式融合和分布式融合两种结构的特点，因而在优缺点方面也是两者的折中；但是，这种结构方式为处理多平台、多个同类/异类传感器的融合问题提供了方便。

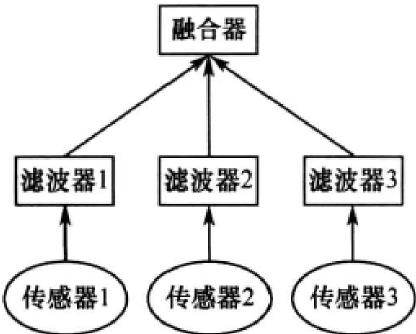


图 11.2　分布式结构

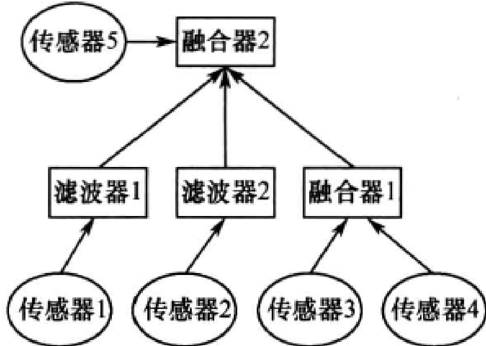


图 11.3　混合式结构

鉴于集中式融合与分布式融合在现实世界中的普遍性和应用的广泛性，接下来主要针对这两种结构进行详细介绍和分析。

11.2　集中式融合滤波算法

考虑以下包含多传感器量测的非线性系统模型

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{w}_k) \tag{11-1}$$

$$\boldsymbol{y}_k^s = h^s(\boldsymbol{x}_k, \boldsymbol{v}_k) \quad s=1,2,\cdots,S \tag{11-2}$$

式中：$\boldsymbol{y}_k^s$ 为第 s 个传感器获取的量测；$h^s(\cdot)$ 为该传感器的观测方程；S 为传感器个数。

若用 $\boldsymbol{y}_{1:k}^{1:S}$ 表示到 k 时刻为止所有传感器获取的累积量测，即 $\boldsymbol{y}_{1:k}^{1:S}=\{\boldsymbol{y}_{1:k}^s\}_{s=1}^S$，$\boldsymbol{y}_{1:k}^s=\{\boldsymbol{y}_i^s\}_{i=1}^k$，则根据贝叶斯原理，$\boldsymbol{x}_k$ 的后验分布 $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}^{1:S})$ 可通过以下两步递推得到。

预测：假定 $k-1$ 时刻状态的后验 PDF 为 $p(\boldsymbol{x}_{k-1} \mid \boldsymbol{y}_{1:k-1})$，则其一步前向预测的 PDF

可表示为

$$p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1}^{1:S}) = \int p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}) p(\boldsymbol{x}_{k-1} \mid \boldsymbol{y}_{1:k-1}^{1:S}) \mathrm{d}\boldsymbol{x}_{k-1} \tag{11-3}$$

式中，隐含假定了 $p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1})$=$p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}, \boldsymbol{y}_{1:k-1})$， $p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1})$ 称为转移密度函数，用于描述式（11-1）中的一阶马尔可夫过程。

滤波：在获得 $p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1}^{1:S})$ 的基础上，综合新近的量测 $\boldsymbol{y}_k^{1:S}$，可通过如下的贝叶斯公式计算 k 时刻状态的后验 PDF：

$$\begin{aligned} p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k}^{1:S}) &= p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1}^{1:S}, \boldsymbol{y}_k^{1:S}) = \frac{p(\boldsymbol{y}_k^{1:S} \mid \boldsymbol{x}_k, \boldsymbol{y}_{1:k-1}^{1:S}) p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1}^{1:S})}{p(\boldsymbol{y}_k^{1:S} \mid \boldsymbol{y}_{1:k-1}^{1:S})} \\ &\propto p(\boldsymbol{y}_k^{1:S} \mid \boldsymbol{x}_k) p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1}^{1:S}) \end{aligned} \tag{11-4}$$

进一步假定各传感器观测独立，于是有

$$\begin{aligned} p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k}^{1:S}) &\propto p(\boldsymbol{y}_k^{1:S} \mid \boldsymbol{x}_k) p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1}^{1:S}) = p(\boldsymbol{y}_k^1, \boldsymbol{y}_k^2, \cdots, \boldsymbol{y}_k^s \mid \boldsymbol{x}_k) p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1}^{1:S}) \\ &= p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1}^{1:S}) \prod_{s=1}^{S} p(\boldsymbol{y}_k^s \mid \boldsymbol{x}_k) \end{aligned} \tag{11-5}$$

式（11-5）即为采用集中式融合结构得到的贝叶斯估计表达式，由此可以看出，对于一般的非线性系统，上式很难得到解析表达式。而在线性、高斯条件下，即若式（11-1）、式（11-2）可以写成

$$\boldsymbol{x}_k = \boldsymbol{F}\boldsymbol{x}_{k-1} + \boldsymbol{w}_k \tag{11-6}$$

$$\boldsymbol{y}_k^s = \boldsymbol{H}^s \boldsymbol{x}_k + \boldsymbol{v}_k^s \quad s=1,2,\cdots,S \tag{11-7}$$

式中：$\boldsymbol{w}_k$ 为零均值高斯过程噪声且 $\mathrm{cov}[\boldsymbol{w}_k]=\boldsymbol{Q}_k$；$\boldsymbol{v}_k^s$ 为零均值高斯观测噪声且 $\mathrm{cov}[\boldsymbol{v}_k^s]=\boldsymbol{R}_k^s$。则可以证明（见附录 F），上述递推式存在最优解，且解的形式为

$$\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{F}\hat{\boldsymbol{x}}_{k-1} \tag{11-8}$$

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{F}\boldsymbol{P}_{k-1}\boldsymbol{F}^{\mathrm{T}} + \boldsymbol{Q}_k \tag{11-9}$$

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k[\boldsymbol{y}_k - \boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1}] \tag{11-10}$$

$$\boldsymbol{P}_k^{-1} = \boldsymbol{P}_{k|k-1}^{-1} + \boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}_k^{-1}\boldsymbol{H} \tag{11-11}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_k\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}_k^{-1} \tag{11-12}$$

式中

$$\boldsymbol{y}_k = [(\boldsymbol{y}_k^1)^{\mathrm{T}} \ (\boldsymbol{y}_k^2)^{\mathrm{T}} \cdots (\boldsymbol{y}_k^S)^{\mathrm{T}}] \tag{11-13}$$

$$\boldsymbol{R}_k^{-1} = \mathrm{diag}[(\boldsymbol{R}_k^1)^{-1} \ (\boldsymbol{R}_k^2)^{-1} \cdots (\boldsymbol{R}_k^S)^{-1}] \tag{11-14}$$

$$\boldsymbol{H} = [(\boldsymbol{H}_k^1)^{\mathrm{T}} \ (\boldsymbol{H}_k^2)^{\mathrm{T}} \cdots (\boldsymbol{H}_k^S)^{\mathrm{T}}]^{\mathrm{T}} \tag{11-15}$$

将式（11-13）～式（11-15）分别代入式（11-10）和式（11-11），得

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{P}_k \sum_{s=1}^{S} (\boldsymbol{H}^s)(\boldsymbol{R}_k^s)^{-1}(\boldsymbol{y}_k^s - \boldsymbol{H}^s\hat{\boldsymbol{x}}_{k|k-1}) \tag{11-16}$$

$$\boldsymbol{P}_k^{-1} = \boldsymbol{P}_{k|k-1}^{-1} + \sum_{s=1}^{S} (\boldsymbol{H}_k^s)^{\mathrm{T}}(\boldsymbol{R}_k^s)^{-1}\boldsymbol{H}_k^s \tag{11-17}$$

需要指出的是，由式（11-5）的表达式可知，上述融合算法主要是针对在同一时刻各传感器的量测噪声之间互不相关的情形。然而在许多实际的应用条件下，这一假设条件不总是满足的（如在共同的噪声环境中对目标运动状态进行观测，其传感器的观测噪

声之间存在一定的相关性，又如即使某些观测误差在原始坐标系中是不相关的，但经非线性坐标变换后由于误差依赖于状态可能变得相关）。对于观测噪声相关的情形，相应的融合算法可参阅文献[6]和文献[7]。

另外，对于一般的非线性系统，式（11-5）的闭式解并不存在，此时需要求次优解。一种可行的方法是用扩展卡尔曼滤波代替卡尔曼滤波算法，此时分别用

$$\boldsymbol{F}_x = \left.\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{x=\hat{x}_{k-1}}, \quad \boldsymbol{H}_x^s = \left.\frac{\partial h^s(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{x=\hat{x}_{k|k-1}}$$

代替 $\boldsymbol{F}$ 及 $\boldsymbol{H}^s$，从而得到基于扩展卡尔曼滤波的集中式融合滤波算法，其流程可概括如下。

Alg（11.1）：EKF based centralized fusion filtering

$$\hat{\boldsymbol{x}}_{k|k-1} = f(\hat{\boldsymbol{x}}_{k-1}) \tag{11-18}$$

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{F}_x \boldsymbol{P}_{k-1} \boldsymbol{F}_x^{\mathrm{T}} + \boldsymbol{Q}_k \tag{11-19}$$

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k[\boldsymbol{y}_k - h(\hat{\boldsymbol{x}}_{k|k-1})] \tag{11-20}$$

$$\boldsymbol{P}_k^{-1} = \boldsymbol{P}_{k|k-1}^{-1} + \boldsymbol{H}^{\mathrm{T}} \boldsymbol{R}_k^{-1} \boldsymbol{H} \tag{11-21}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_k \boldsymbol{H}^{\mathrm{T}} \boldsymbol{R}_k^{-1} \tag{11-22}$$

式中：$\boldsymbol{y}_k$ 与 $\boldsymbol{R}_k$ 的定义与式（11-13）和式（11-14）中的相同，且

$$\boldsymbol{H} = [(\boldsymbol{H}_x^1)^{\mathrm{T}} \ (\boldsymbol{H}_x^2)^{\mathrm{T}} \cdots (\boldsymbol{H}_x^S)^{\mathrm{T}}]^{\mathrm{T}} \tag{11-23}$$

$$\boldsymbol{h}(\cdot) = [(h^1(\cdot))^{\mathrm{T}} \ (h^2(\cdot))^{\mathrm{T}} \cdots (h^S(\cdot))^{\mathrm{T}}]^{\mathrm{T}} \tag{11-24}$$

对于本书中第二部分提到的其他类型的高斯近似非线性滤波方法（如 UKF、CDKF、GHKF 及 CKF 等），由于不存在显式的 $\boldsymbol{F}_x$ 和 $\boldsymbol{H}_x$，不能作类似的处理。但由于式（11-11）和式（11-12）可改写成

$$\boldsymbol{P}_k = \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k \boldsymbol{P}_{y_k y_k} \boldsymbol{K}_k^{-1} \tag{11-25}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_{x_k y_k} \boldsymbol{P}_{y_k y_k}^{-1} \tag{11-26}$$

因此只要通过其他方式求得 $\boldsymbol{P}_{x_k y_k}$ 和 $\boldsymbol{P}_{y_k y_k}^{-1}$，同样可以实现相应的融合滤波处理，在此不再赘述。

上述介绍的是在卡尔曼滤波框架下的融合算法，在粒子滤波框架下，状态的联合后验分布 $p(\boldsymbol{x}_{1:k} \mid \boldsymbol{y}_{1:k}^{1:S})$（$p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k}^{1:S})$ 为 $p(\boldsymbol{x}_{1:k} \mid \boldsymbol{y}_{1:k}^{1:S})$ 的边缘化形式）被表示成一组随机采样 $\{\boldsymbol{x}_{1:k}^{(i)}\}_{i=1}^N$ 的形式，且各采样样本的重要性权值为

$$w_k^{(i)} \propto \frac{p(\boldsymbol{x}_{1:k}^{(i)} \mid \boldsymbol{y}_{1:k}^{1:S})}{q(\boldsymbol{x}_{1:k}^{(i)} \mid \boldsymbol{y}_{1:k}^{1:S})} \tag{11-27}$$

式中：$p(\boldsymbol{x}_{1:k} \mid \boldsymbol{y}_{1:k}^{1:S})$ 根据贝叶斯定理可进一步写成

$$\begin{aligned} p(\boldsymbol{x}_{1:k} \mid \boldsymbol{y}_{1:k}^{1:S}) &= \frac{p(\boldsymbol{y}_k^{1:S} | \boldsymbol{x}_k) p(\boldsymbol{x}_k | \boldsymbol{x}_{k-1})}{p(\boldsymbol{y}_k^{1:S} | \boldsymbol{y}_{1:k-1}^{1:S})} p(\boldsymbol{x}_{1:k-1} \mid \boldsymbol{y}_{1:k-1}^{1:S}) \\ &\propto p(\boldsymbol{y}_k^{1:S} | \boldsymbol{x}_k) p(\boldsymbol{x}_k | \boldsymbol{x}_{k-1}) p(\boldsymbol{x}_{1:k-1} \mid \boldsymbol{y}_{1:k-1}^{1:S}) \end{aligned} \tag{11-28}$$

若将重要权定义为满足如下的因式分解形式，即

$$q(\boldsymbol{x}_{1:k} \mid \boldsymbol{y}_{1:k}^{1:S}) \triangleq q(\boldsymbol{x}_k | \boldsymbol{x}_{1:k-1}, \boldsymbol{y}_{1:k}^{1:S}) q(\boldsymbol{x}_{1:k-1} \mid \boldsymbol{y}_{1:k-1}^{1:S}) \tag{11-29}$$

则权值的递推式满足

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(\boldsymbol{y}_k^{1:S}|\boldsymbol{x}_k^{(i)})p(\boldsymbol{x}_k^{(i)}|\boldsymbol{x}_{k-1}^{(i)})}{q(\boldsymbol{x}_k^{(i)}|\boldsymbol{x}_{1:k-1}^{(i)},\boldsymbol{y}_{1:k}^{1:S})} \tag{11-30}$$

同样，若假设各传感器之间的观测噪声相互独立，则式（11-30）等价于

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(\boldsymbol{x}_k^{(i)}|\boldsymbol{x}_{k-1}^{(i)})\prod_{s=1}^{S} p(\boldsymbol{y}_k^s|\boldsymbol{x}_k^{(i)})}{q(\boldsymbol{x}_k^{(i)}|\boldsymbol{x}_{1:k-1}^{(i)},\boldsymbol{y}_{1:k}^{1:S})} \tag{11-31}$$

式中，与单传感器观测条件下的重要性权值计算类似，多传感器下的重要性权值计算同样依赖于重要性采样函数 $q(\boldsymbol{x}_k|\boldsymbol{x}_{1:k-1},\boldsymbol{y}_{1:k}^{1:S})$ 选择，当 $q(\boldsymbol{x}_k|\boldsymbol{x}_{1:k-1},\boldsymbol{y}_{1:k}^{1:S})=p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ 时，式（11-31）可进一步简化为

$$w_k^{(i)} \propto w_{k-1}^{(i)} \prod_{s=1}^{S} p(\boldsymbol{y}_k^s|\boldsymbol{x}_k^{(i)}) \tag{11-32}$$

对于其他的选择方式，同样可以假定最优重要权满足

$$\begin{aligned} q(\boldsymbol{x}_k|\boldsymbol{x}_{1:k-1},\boldsymbol{y}_{1:k}^{1:S})&=q(\boldsymbol{x}_k|\boldsymbol{x}_{1:k-1},\boldsymbol{y}_k^{1:S})=\prod_{s=1}^{S} q(\boldsymbol{x}_k|\boldsymbol{x}_{1:k-1},\boldsymbol{y}_k^s) \\ &\approx \prod_{s=1}^{S} \mathcal{N}(\boldsymbol{x}_k;\hat{\boldsymbol{x}}_k^s,P_k^s) \end{aligned} \tag{11-33}$$

对于其他更一般的情况，其计算过程要比上述复杂，如对于观测噪声相关的情形，可采用主分量（PCA）[8]等去相关的手段将其变换成非相关噪声，进而采用相应的方法来处理，具体过程可参阅文献[9]。

11.3 分布式融合滤波算法

在如图 11.2 所示的分布式融合结构中，对来自不同传感器航迹（传播到同一时刻）的状态估计进行关联和融合以得到系统航迹的状态估计。在这一过程中，不利用系统航迹以前的状态估计。因此，对于每一个传感器，其从量测到状态的估计仍可通过第 1 章所介绍的贝叶斯公式进行计算，即对第 s 个传感器有

$$p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1}^s) = \int p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}) p(\boldsymbol{x}_{k-1} \mid \boldsymbol{y}_{1:k-1}^s) \mathrm{d}\boldsymbol{x}_{k-1} \tag{11-34}$$

$$p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k}^s) = p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1}^s, \boldsymbol{y}_k^s) \propto p(\boldsymbol{y}_k^s \mid \boldsymbol{x}_k) p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1}^s) \tag{11-35}$$

根据特定的非线性系统状态，对上述两式进行求解可得到各种次优解，即得到各个传感器基于量测值的非线性局部估计（可通过第二部分和第三部分介绍的方法得到）。分布式融合的目的是根据这些局部估计得到全局估计，根据不同的条件，可得到不同的融合策略与规则[3]，其中有些融合方法是最优的、有的是次优的。这里仅对几种最有代表性的融合算法进行详细介绍。

由系统模型式（11-6）和式（11-7）可知，对于分布式融合结构，每个局部传感器 s 的最优估计结果可以用信息滤波的形式表示为

$$\hat{\boldsymbol{x}}_k^s = \hat{\boldsymbol{x}}_{k|k-1}^s + \boldsymbol{P}_k^s (\boldsymbol{H}^s)^{\mathrm{T}} (\boldsymbol{R}_k^s)^{-1} (\boldsymbol{y}_k^s - \boldsymbol{H}^s \hat{\boldsymbol{x}}_{k|k-1}^s) \tag{11-36}$$

$$(\boldsymbol{P}_k^s)^{-1} = (\boldsymbol{P}_{k|k-1}^s)^{-1} + (\boldsymbol{H}_k^s)^{\mathrm{T}} (\boldsymbol{R}_k^s)^{-1} \boldsymbol{H}_k^s \tag{11-37}$$

在分布式融合结构中，中心融合器将不直接使用量测来获取全局估计，而仅仅联合各传感器的局部估计来得到全局估计。为此首先将式（11-36）和式（11-37）左右两边对应相乘，得

$$
\begin{aligned}
(\boldsymbol{P}_k^s)^{-1}\hat{\boldsymbol{x}}_k^s &= [(\boldsymbol{P}_{k|k-1}^s)^{-1}+(\boldsymbol{H}_k^s)^{\mathrm{T}}(\boldsymbol{R}_k^s)^{-1}\boldsymbol{H}_k^s] \\
&\quad \cdot[\hat{\boldsymbol{x}}_{k|k-1}^s+\boldsymbol{P}_k^s(\boldsymbol{H}^s)^{\mathrm{T}}(\boldsymbol{R}_k^s)^{-1}(\boldsymbol{y}_k^s-\boldsymbol{H}^s\hat{\boldsymbol{x}}_{k|k-1}^s)] \\
&= [(\boldsymbol{P}_{k|k-1}^s)^{-1}+(\boldsymbol{H}_k^s)^{\mathrm{T}}(\boldsymbol{R}_k^s)^{-1}\boldsymbol{H}_k^s]\hat{\boldsymbol{x}}_{k|k-1}^s \\
&\quad +(\boldsymbol{P}_k^i)^{-1}\boldsymbol{P}_k^s(\boldsymbol{H}^s)^{\mathrm{T}}(\boldsymbol{R}_k^s)^{-1}(\boldsymbol{y}_k^s-\boldsymbol{H}^s\hat{\boldsymbol{x}}_{k|k-1}^s) \\
&= (\boldsymbol{P}_{k|k-1}^s)^{-1}\hat{\boldsymbol{x}}_{k|k-1}^s+(\boldsymbol{H}_k^s)^{\mathrm{T}}(\boldsymbol{R}_k^s)^{-1}\boldsymbol{y}_k^s
\end{aligned}
\tag{11-38}
$$

于是有

$$
(\boldsymbol{H}_k^s)^{\mathrm{T}}(\boldsymbol{R}_k^s)^{-1}\boldsymbol{y}_k^s=(\boldsymbol{P}_k^s)^{-1}\hat{\boldsymbol{x}}_k^s-(\boldsymbol{P}_{k|k-1}^s)^{-1}\hat{\boldsymbol{x}}_{k|k-1}^s \tag{11-39}
$$

接下来，将用上式来消除融合中心更新方程中的量测信息。在此之前首先根据集中式融合的结果

$$
\hat{\boldsymbol{x}}_k=\hat{\boldsymbol{x}}_{k|k-1}+\boldsymbol{P}_k\sum_{s=1}^{S}(\boldsymbol{H}^s)(\boldsymbol{R}_k^s)^{-1}(\boldsymbol{y}_k^s-\boldsymbol{H}^s\hat{\boldsymbol{x}}_{k|k-1}) \tag{11-40}
$$

$$
\boldsymbol{P}_k^{-1}=\boldsymbol{P}_{k|k-1}^{-1}+\sum_{s=1}^{S}(\boldsymbol{H}_k^s)^{\mathrm{T}}(\boldsymbol{R}_k^s)^{-1}\boldsymbol{H}_k^s \tag{11-41}
$$

对式（11-40）和式（11-41）采用与前述类似的等号两边相乘的方法，得

$$
\boldsymbol{P}_k^{-1}\hat{\boldsymbol{x}}_k=\boldsymbol{P}_{k|k-1}^{-1}\hat{\boldsymbol{x}}_{k|k-1}+\sum_{s=1}^{S}(\boldsymbol{H}^s)(\boldsymbol{R}_k^s)^{-1}\boldsymbol{y}_k^s \tag{11-42}
$$

将式（11-39）代入式（11-42），得

$$
\boldsymbol{P}_k^{-1}\hat{\boldsymbol{x}}_k=\boldsymbol{P}_{k|k-1}^{-1}\hat{\boldsymbol{x}}_{k|k-1}+\sum_{s=1}^{S}[(\boldsymbol{P}_k^s)^{-1}\hat{\boldsymbol{x}}_k^s-(\boldsymbol{P}_{k|k-1}^s)^{-1}\hat{\boldsymbol{x}}_{k|k-1}^s] \tag{11-43}
$$

同理，将式（11-37）变形，得

$$
(\boldsymbol{H}_k^s)^{\mathrm{T}}(\boldsymbol{R}_k^s)^{-1}\boldsymbol{H}_k^s=(\boldsymbol{P}_k^s)^{-1}-(\boldsymbol{P}_{k|k-1}^s)^{-1} \tag{11-44}
$$

将式（11-44）代入式（11-41），得

$$
\boldsymbol{P}_k^{-1}=\boldsymbol{P}_{k|k-1}^{-1}+\sum_{s=1}^{S}[(\boldsymbol{P}_k^s)^{-1}-(\boldsymbol{P}_{k|k-1}^s)^{-1}] \tag{11-45}
$$

综上，式（11-43）和式（11-45）即为采用分布式结构得到的融合结果，由推导过程可以看出，该融合算法是通过具有全局最优的集中式融合算法通过矩阵变换得到的，因而也是全局最优的。

需要指出的是，上述融合算法中，融合中心不仅需要各传感器的滤波估计 $\{\hat{\boldsymbol{x}}_k^s,\boldsymbol{P}_k^s\}_{s=1}^{S}$，同时还需要各自的预测估计 $\{\hat{\boldsymbol{x}}_{k|k-1}^s,\boldsymbol{P}_{k|k-1}^s\}_{s=1}^{S}$，当然这些信息对一般的序贯滤波算法而言都能得到。从式（11-43）和式（11-45）可以看出，这种融合算法不依赖于单个传感器具体的滤波算法形式，为各种非线性滤波算法的应用提供了方便。

但同样需要强调的是，上述结果是在线性、高斯条件下导出的，对于一般的非线性系统，上述融合算法在具体应用时需根据不同的滤波方法进行修正。特别地，当采用 EKF 算法时，$\boldsymbol{H}_k^s$ 对应于各节点传感器观测方程的偏导数。而对于 UKF、CDKF 或 CKF 等其

他滤波算法，则需要进行特殊的处理才能实现融合。下面以文献[10]中的不敏信息滤波算法为例，介绍基于 UKF 的融合过程。

对于下式所描述的系统方程

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{w}_k) \tag{11-46}$$

$$\boldsymbol{y}_k = h(\boldsymbol{x}_k) + \boldsymbol{v}_k \tag{11-47}$$

若定义 Fisher 信息矩阵 $\boldsymbol{J}_k = \boldsymbol{P}_k^{-1}$ 及信息状态矢量 $\boldsymbol{z}_k$ 分别为

$$\boldsymbol{Z}_k \triangleq \boldsymbol{P}_k^{-1} = \boldsymbol{J}_k \tag{11-48}$$

$$\hat{\boldsymbol{z}}_k \triangleq \boldsymbol{P}_k^{-1}\hat{\boldsymbol{x}}_k = \boldsymbol{Z}_k\hat{\boldsymbol{x}}_k \tag{11-49}$$

根据扩展信息滤波（EIF）原理，上述两式的更新方程为

$$\boldsymbol{Z}_k = \boldsymbol{P}_{k|k-1}^{-1} + \boldsymbol{H}_k^{\mathrm{T}}\boldsymbol{R}_k^{-1}\boldsymbol{H}_k = \boldsymbol{Z}_{k|k-1} + \boldsymbol{I}_k \tag{11-50}$$

$$\hat{\boldsymbol{z}}_k = \hat{\boldsymbol{z}}_{k|k-1} + \boldsymbol{H}_k^{\mathrm{T}}\boldsymbol{R}_k^{-1}[\boldsymbol{v}_k + \boldsymbol{H}_k\hat{\boldsymbol{x}}_{k|k-1}] = \hat{\boldsymbol{z}}_{k|k-1} + \boldsymbol{i}_k \tag{11-51}$$

式中：$\boldsymbol{i}_k$ 为信息状态贡献量；$\boldsymbol{I}_k$ 为 $\boldsymbol{i}_k$ 的协方差矩阵。其定义为

$$\boldsymbol{i}_k = \boldsymbol{H}_k^{\mathrm{T}}\boldsymbol{R}_k^{-1}[\boldsymbol{v}_k + \boldsymbol{H}_k\hat{\boldsymbol{x}}_{k|k-1}] \tag{11-52}$$

$$\boldsymbol{I}_k = \boldsymbol{H}_k^{\mathrm{T}}\boldsymbol{R}_k^{-1}\boldsymbol{H}_k \tag{11-53}$$

式中：$\boldsymbol{H}_k$ 为观测方程的偏导数；$\boldsymbol{v}_k$ 为新息（innovation），且 $\boldsymbol{v}_k = \boldsymbol{y}_k - h(\hat{\boldsymbol{x}}_{k|k-1})$。

由此可得预测信息状态及协方差为

$$\hat{\boldsymbol{z}}_{k|k-1} = \boldsymbol{P}_{k|k-1}^{-1}\hat{\boldsymbol{x}}_{k|k-1} \tag{11-54}$$

$$\boldsymbol{Z}_{k|k-1} = \boldsymbol{P}_{k|k-1}^{-1} = [\boldsymbol{F}_k\boldsymbol{P}_{k-1}\boldsymbol{F}_k^{\mathrm{T}} + \boldsymbol{Q}_k]^{-1} \tag{11-55}$$

式中：$\boldsymbol{F}_k$ 为状态方程的偏导数。

当采用 UKF 算法时，首先定义扩展状态及协方差矩阵

$$\hat{\boldsymbol{x}}_k^a = \begin{bmatrix} \hat{\boldsymbol{x}}_k \\ \hat{\boldsymbol{w}}_k \end{bmatrix}, \quad \boldsymbol{P}_k^a = \begin{bmatrix} \boldsymbol{P}_k & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}_k \end{bmatrix}$$

由此得到 $2n_a+1$ 个样点

$$\chi_{0,k}^a = \hat{\boldsymbol{x}}_k^a$$

$$\chi_{i,k}^a = \hat{\boldsymbol{x}}_k^a + \left(\sqrt{(n_a+\lambda)\boldsymbol{P}_k^a}\right)_i$$

$$\chi_{i,k}^a = \hat{\boldsymbol{x}}_k^a - \left(\sqrt{(n_a+\lambda)\boldsymbol{P}_k^a}\right)_i$$

式中：$\lambda = \alpha^2(n_a+\kappa) - n_a$，$n_a$ 为扩展状态的维数，其他参数的定义可参阅第 4 章。

样点的权值分配为

$$\begin{cases} w_0^{(m)} = \lambda/(\lambda+n_a) \\ w_i^{(m)} = 1/(2\lambda+2n_a) \end{cases} \quad (i=1,\cdots,2n_a)$$

$$\begin{cases} w_0^{(c)} = \lambda/(\lambda+n_a) + (1-\alpha^2+\beta) \\ w_i^{(c)} = 1/(2\lambda+2n_a) \end{cases} \quad (i=1,\cdots,2n_a)$$

由此，利用 UT 变换可得信息预测方程为

$$\hat{\boldsymbol{z}}_{k+1|k} = \boldsymbol{Z}_{k+1|k}\sum_{i=0}^{2n_a} w_i^{(m)}\chi_{i,k+1|k}^x \tag{11-56}$$

$$\boldsymbol{Z}_{k+1|k}=\boldsymbol{P}_{k+1|k}^{-1} \tag{11-57}$$

式中：$\chi_{i,k+1}^{x}$ 为状态的预测样点；$\boldsymbol{P}_{k|k-1}$ 为相应的协方差矩阵，且有

$$\chi_{i,k+1|k}^{x}=f\left(\chi_{i,k}^{x},\chi_{i,k}^{w}\right) \tag{11-58}$$

$$\boldsymbol{P}_{k+1|k}=\sum_{i=0}^{2n_a}w_i^{(c)}\left[\chi_{i,k+1|k}^{x}-\hat{\boldsymbol{x}}_{k+1|k}\right]\left[\chi_{i,k+1|k}^{x}-\hat{\boldsymbol{x}}_{k+1|k}\right]^{\mathrm{T}} \tag{11-59}$$

在 EKF 算法中，观测的预测误差自协方差和互协方差分别为

$$\boldsymbol{P}_{yy,k+1}=E[(\boldsymbol{y}_{k+1}-\hat{\boldsymbol{y}}_{k+1|k})(\boldsymbol{y}_{k+1}-\hat{\boldsymbol{y}}_{k+1|k})^{\mathrm{T}}]\approx\boldsymbol{H}_{k+1}\boldsymbol{P}_{k+1|k}\boldsymbol{H}_{k+1}^{\mathrm{T}} \tag{11-60}$$

$$\boldsymbol{P}_{xy,k+1}=E[(\boldsymbol{x}_{k+1}-\hat{\boldsymbol{x}}_{k+1|k})(\boldsymbol{y}_{k+1}-\hat{\boldsymbol{y}}_{k+1|k})^{\mathrm{T}}]\approx\boldsymbol{P}_{k+1|k}\boldsymbol{H}_{k+1}^{\mathrm{T}} \tag{11-61}$$

通过类比，利用式（11-61）可将式（11-53）表示为

$$\begin{aligned}\boldsymbol{I}_{k+1}&=\boldsymbol{H}_{k+1}^{\mathrm{T}}\boldsymbol{R}_{k+1}^{-1}\boldsymbol{H}_{k+1}=\boldsymbol{P}_{k+1|k}^{-1}\boldsymbol{P}_{k+1|k}\boldsymbol{H}_{k+1}^{\mathrm{T}}\boldsymbol{R}_{k+1}^{-1}\boldsymbol{H}_{k+1}\boldsymbol{P}_{k+1|k}^{\mathrm{T}}(\boldsymbol{P}_{k+1|k}^{\mathrm{T}})^{-1}\\&=\boldsymbol{P}_{k+1|k}^{-1}\boldsymbol{P}_{xy,k+1}\boldsymbol{R}_{k+1}^{-1}(\boldsymbol{P}_{xy,k+1})^{\mathrm{T}}(\boldsymbol{P}_{k+1|k}^{\mathrm{T}})^{-1}\end{aligned} \tag{11-62}$$

式中：$\boldsymbol{P}_{k+1|k}$ 可通过式（11-59）计算得到；而 $\boldsymbol{P}_{xy,k+1}$ 则可通过 UT 计算为

$$\boldsymbol{P}_{xy,k+1}=\sum_{i=0}^{2n_a}w_i^{(c)}[\chi_{i,k+1|k}^{x}-\hat{\boldsymbol{x}}_{k+1|k}][\mathcal{Y}_{i,k+1|k}-\hat{\boldsymbol{y}}_{k+1|k}]^{\mathrm{T}} \tag{11-63}$$

式中

$$\mathcal{Y}_{i,k+1|k}=h\left(\chi_{i,k+1|k}^{x}\right),\quad \hat{\boldsymbol{y}}_{k+1|k}=\sum_{i=0}^{2n_a}w_i^{(m)}\mathcal{Y}_{i,k+1|k}$$

同理，式（11-52）中的 $\boldsymbol{i}_{k+1}$ 可计算为

$$\begin{aligned}\boldsymbol{i}_{k+1}&=\boldsymbol{H}_{k+1}^{\mathrm{T}}\boldsymbol{R}_{k+1}^{-1}[\boldsymbol{v}_{k+1}+\boldsymbol{H}_{k+1}\hat{\boldsymbol{x}}_{k+1|k}]\\&=\boldsymbol{P}_{k+1|k}^{-1}\boldsymbol{P}_{k+1|k}\boldsymbol{H}_{k+1}^{\mathrm{T}}\boldsymbol{R}_{k+1}^{-1}[\boldsymbol{v}_{k+1}+\boldsymbol{H}_{k+1}(\boldsymbol{P}_{k+1|k})^{\mathrm{T}}(\boldsymbol{P}_{k+1|k}^{\mathrm{T}})^{-1}\hat{\boldsymbol{x}}_{k+1|k}]\\&=\boldsymbol{P}_{k+1|k}^{-1}\boldsymbol{P}_{xy,k+1}\boldsymbol{R}_{k+1}^{-1}[\boldsymbol{v}_{k+1}+(\boldsymbol{P}_{xy,k+1})^{\mathrm{T}}(\boldsymbol{P}_{k+1|k}^{\mathrm{T}})^{-1}\hat{\boldsymbol{x}}_{k+1|k}]\end{aligned} \tag{11-64}$$

为了能利用 EIF 的算法框架，可定义伪观测矩阵

$$\mathcal{H}_{k+1}^{\mathrm{T}}\triangleq\boldsymbol{P}_{k+1|k}^{-1}\boldsymbol{P}_{xy,k+1} \tag{11-65}$$

由此可得信息贡献量的更新式为

$$\boldsymbol{i}_{k+1}=\mathcal{H}_{k+1}^{\mathrm{T}}\boldsymbol{R}_{k+1}^{-1}[\boldsymbol{v}_{k+1}+\mathcal{H}_{k+1}\hat{\boldsymbol{x}}_{k+1|k}] \tag{11-66}$$

$$\boldsymbol{I}_{k+1}=\mathcal{H}_{k+1}^{\mathrm{T}}\boldsymbol{R}_{k+1}^{-1}\mathcal{H}_{k+1} \tag{11-67}$$

及系统状态的更新

$$\hat{\boldsymbol{x}}_k=\boldsymbol{P}_k\hat{\boldsymbol{z}}_k \tag{11-68}$$

在多传感器观测条件下，第 s 个 $(s=1,2,\cdots,S)$ 传感器的信息贡献量为

$$\boldsymbol{i}_{k+1}^{s}=\mathcal{H}_{k+1}^{s}(\boldsymbol{R}_{k+1}^{s})^{-1}[\boldsymbol{v}_{k+1}^{s}+\mathcal{H}_{k+1}^{s}\hat{\boldsymbol{x}}_{k+1|k}] \tag{11-69}$$

$$\boldsymbol{I}_{k+1}^{s}=(\mathcal{H}_{k+1}^{s})^{\mathrm{T}}(\boldsymbol{R}_{k+1}^{s})^{-1}\mathcal{H}_{k+1}^{s} \tag{11-70}$$

由此可得多传感器的融合结果为

$$\hat{\boldsymbol{z}}_{k+1}=\hat{\boldsymbol{z}}_{k+1|k}+\sum_{s=1}^{S}\boldsymbol{i}_{k+1}^{s} \tag{11-71}$$

$$\boldsymbol{Z}_{k+1}=\boldsymbol{Z}_{k|k-1}+\sum_{s=1}^{S}\boldsymbol{I}_{k+1}^{s} \tag{11-72}$$

式中：$\hat{\boldsymbol{z}}_{k+1}$、$\boldsymbol{Z}_{k|k-1}$ 可通过式（11-56）和式（11-57）得到。

式（11-69）～式（11-72）构成了不敏信息融合（UIF）的递推过程，对于其他滤波算法如 CDKF、CKF，可采用完全类似的融合方法。另外，在融合策略方面，除了采用如上所述的状态融合外，也可采用粒子融合[11, 12]的方法，由于这方面的理论相对不够成熟，在此不再细述。

11.4 应用实例

11.4.1 多红外传感器融合跟踪

考虑二维平面内的只测角跟踪问题，目标的运动方程满足

$$\boldsymbol{x}_{k+1}=\boldsymbol{F}\boldsymbol{x}_k+\boldsymbol{w}_k \tag{11-73}$$

式中：状态 $\boldsymbol{x}_k=[x_k\ \dot{x}_k\ y_k\ \dot{y}_k]^{\mathrm{T}}$；矩阵 $\boldsymbol{F}=\mathrm{diag}\{\boldsymbol{F}_1,\boldsymbol{F}_1\}$；$T$ 为观测时间间隔；$\boldsymbol{w}_k$ 为零均值扰动噪声，其协方差阵 $\boldsymbol{Q}_k=\mathrm{diag}\{\boldsymbol{Q}_1,\boldsymbol{Q}_1\}$。另外，$\boldsymbol{F}_1=\begin{bmatrix}1 & T\\ 0 & 1\end{bmatrix}$，$\boldsymbol{Q}_1=q^2\begin{bmatrix}T^4/3 & T^3/2\\ T^3/2 & T^2\end{bmatrix}$。

采用位于不同位置的 S 个红外传感器进行观测，第 s 个传感器的观测方程为

$$\varphi_k^s=\arctan\left[\frac{y_k-y_k^s}{x_k-x_k^s}\right]+v_k^s \tag{11-74}$$

式中：(x_k^s,y_k^s) 表示 k 时刻传感器 $s(s=1,2,\cdots,S)$ 的位置；v_k^s 为零均值高斯噪声，即 $v_k^s\sim\mathcal{N}(v;0,\sigma_v^2)$，且假设不同传感器间的观测相互独立。

仿真中假定目标初始状态 $\boldsymbol{x}_0=[20\text{km}\ \ -200\text{m/s}\ 12\text{km}\ 100\text{m/s}]^{\mathrm{T}}$，$T$=0.5s，$q$=2m/s^2。采用 3 个固定位置的传感器进行观测（传感器的位置如图 11.4 所示），观测精度 σ_v=2mrad。进一步假定初始估计误差协方差矩阵 $\boldsymbol{P}_0$=diag([250^2 50^2 250^2 50^2])，L=2km，采用 EKF 和 UKF 集中式融合方法进行跟踪，实验次数为 500，所得结果如图 11.5～图 11.9 所示。

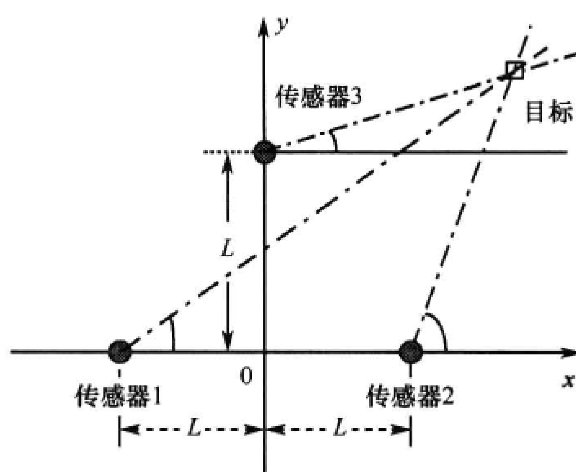


图 11.4 传感器配置

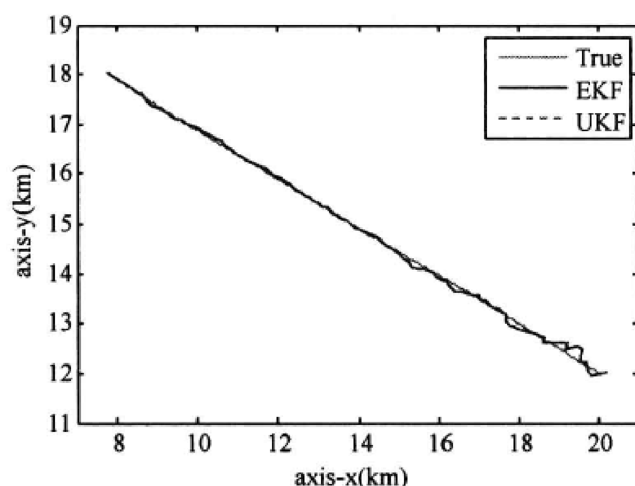


图 11.5 单次跟踪示例

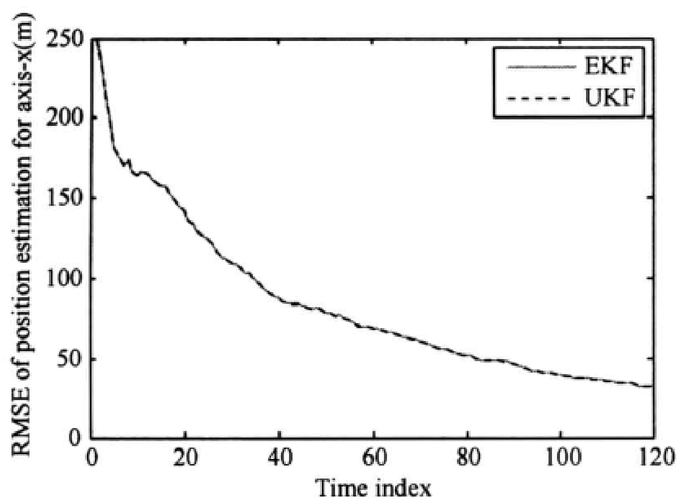


图 11.6　*x* 轴位置估计均方根误差

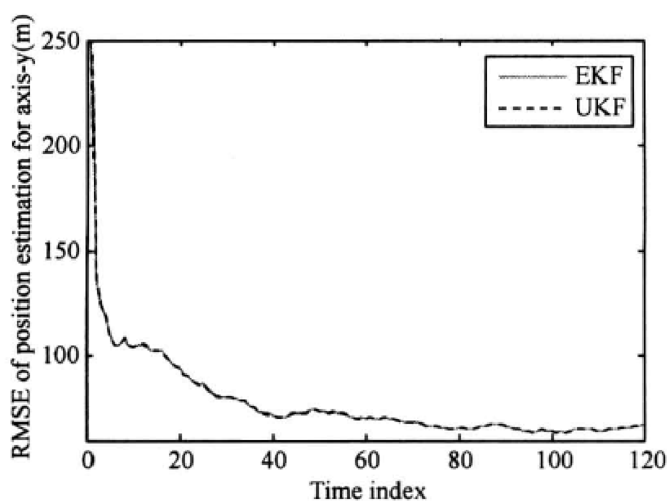


图 11.7　*y* 轴位置估计均方根误差

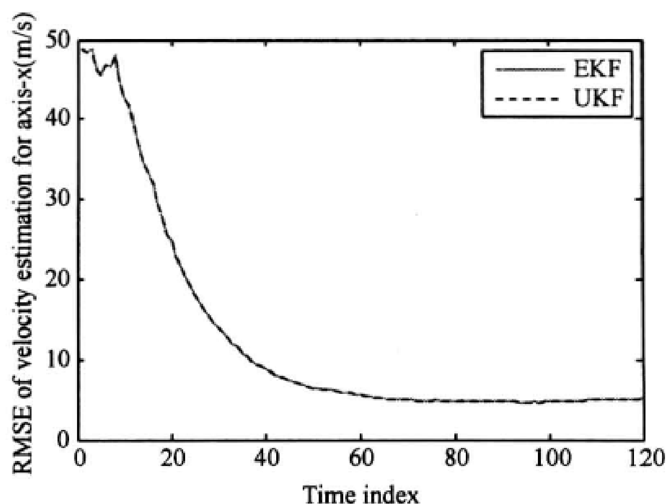


图 11.8　*x* 轴速度估计均方根误差

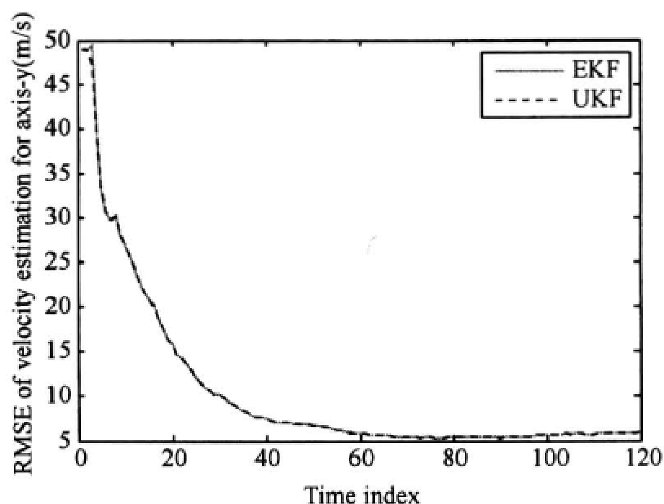


图 11.9　*y* 轴速度估计均方根误差

由图 11.5～图 11.9 中的结果可以看出，采用 3 个传感器观测，进而利用 EKF 和 UKF 进行融合跟踪，可以得到良好的估计效果，说明了集中式融合方法的有效性。另外，从图中的结果看，两种算法的性能非常接近，说明在处理这类问题时具有等效性。对于只测角跟踪问题，由于存在可观测性条件的限制，这里没有给出单传感器的跟踪结果。

为了进一步考察传感器间距对跟踪误差的影响，表 11.1 比较了 L=0.5km,1km, 2km,4km 等不同条件下的 ARMSE（大于 10s）。由此可以看出，随着传感器间距的不断增加，跟踪精度也不断提高；究其原因，是因为当 L 较大时，具有更好的可观测条件，从而降低了估计误差。

表 11.1 不同传感器配置条件下的跟踪结果

L \ ARMSE	pos_x/m	pos_y/m	vel_x/（m/s）	vel_y/（m/s）
L=500m	200.72	224.70	11.18	12.65
L=1000m	126.83	141.94	9.49	9.95
L=2000m	65.64	70.49	7.14	6.74
L=4000m	37.24	36.55	5.73	5.11

11.4.2 多雷达传感器融合跟踪

考虑二维平面内的距离扩展目标跟踪问题，假定目标运动满足

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}\boldsymbol{x}_k + \boldsymbol{w}_k \tag{11-75}$$

式中：状态 $\boldsymbol{x}_k = [x_k\ \dot{x}_k\ y_k\ \dot{y}_k\ \ell_k]^{\mathrm{T}}$；矩阵 $\boldsymbol{F} = \mathrm{diag}\{\boldsymbol{F}_1, \boldsymbol{F}_1, 1\}$；$T$ 为观测间隔；$\boldsymbol{w}_k$ 为零均值过程噪声，其协方差阵 $\boldsymbol{Q}_k = \mathrm{diag}\{\boldsymbol{Q}_1, \boldsymbol{Q}_1, (Tq_1)^2\}$。另外，$\boldsymbol{F}_1 = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$，$\boldsymbol{Q}_1 = q^2 \begin{bmatrix} T^4/3 & T^3/2 \\ T^3/2 & T^2 \end{bmatrix}$。

采用 S 个雷达传感器进行观测，第 s 个传感器的观测方程为

$$\boldsymbol{z}_k^s = \begin{bmatrix} r_k^s \\ \varphi_k^s \\ l_k^s \end{bmatrix} = \boldsymbol{h}(\boldsymbol{x}_k^s) + \boldsymbol{v}_k^s = \begin{bmatrix} \sqrt{(x_k^s)^2 + (y_k^s)^2} \\ \arctan(y_k^s / x_k^s) \\ \ell \left|\cos\phi_k^s\right| \end{bmatrix} + \begin{bmatrix} v_{r,k}^s \\ v_{\varphi,k}^s \\ v_{l,k}^s \end{bmatrix} \tag{11-76}$$

式中：$\boldsymbol{x}_k^s = \boldsymbol{x}_k - \boldsymbol{x}_{s,k}$ 为 k 时刻目标（$\boldsymbol{x}_k$）与第 s 个传感器（$\boldsymbol{x}_{s,k}$）的相对运动状态，且

$$\phi_k = \arccos\frac{x_k \dot{x}_k + y_k \dot{y}_k}{\sqrt{x_k^2 + y_k^2}\sqrt{\dot{x}_k^2 + \dot{y}_k^2}} \tag{11-77}$$

ℓ 为目标真实的径向长度；$v_{r,k}^s$、$v_{\varphi,k}^s$ 和 $v_{l,k}^s$ 分别为目标距离、方位和径向长度观测噪声，且都满足零均值高斯分布。

仿真中假定目标初始状态 $\boldsymbol{x}_0 = [16\mathrm{km}\ 16\mathrm{m/s}\ 8\mathrm{km}\ 12\mathrm{m/s}\ 60\mathrm{m}]^{\mathrm{T}}$，初始估计误差协方差矩阵 $\boldsymbol{P}_0 = \mathrm{diag}([100^2\ \ 20^2\ \ 100^2\ \ 20^2\ \ 5^2])$，$T$=0.5s，$q$=1m/s^2，$q_1$=0.02m。采用 3 个固定的传感器进行观测，其位置分别为 $(-2\mathrm{km},0)$、$(0,2\mathrm{km})$ 和 $(2\mathrm{km},0)$，距离、方位和径向长度的观测精度分别为 10m、0.3° 和 2m。采用式（11-32）所示的集中式融合粒子滤波算法进行跟踪，粒子数目 M=6000，初始粒子集通过 $\boldsymbol{P}_0$ 产生，实验次数为 100，所得结果如图 11.10～图 11.15 所示。作为对比，图中同时给出了位于原点的单传感器跟踪结果。

由图中的结果可以看出，通过粒子滤波算法可实现对距离扩展目标的有效估计，且因为粒子具有散布性特点，使其能在很短的时间内迅速达到稳定的跟踪状态。相比之下，多传感器融合估计的精度比单传感器要高，尤其是对于目标位置估计。

为了进一步比较粒子数目对估计性能的影响，表 11.2 给出了不同粒子数目条件下跟踪接近稳定后（大于 10s）的 ARMSE。可此可以看出，随着粒子数目的增加，估计性能也相应地有了不同程度的改善；但当粒子数目大到一定程度后（如 M>6000），增加粒子数目对跟踪性能的改善不再明显。但不论在哪种条件下，多传感器跟踪精度要一致高于单传感器，说明了多传感器融合跟踪的优势。

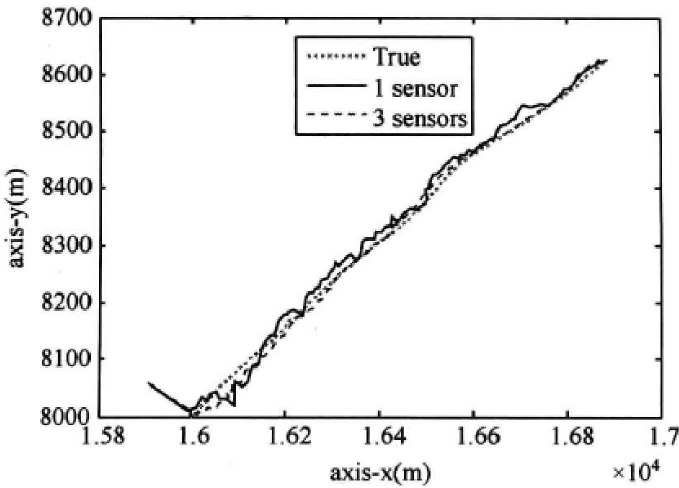


图 11.10 单次跟踪示例

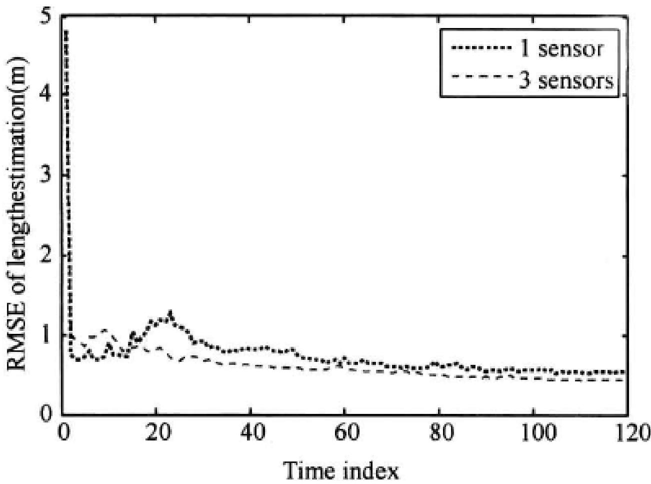


图 11.11 长度估计均方根误差

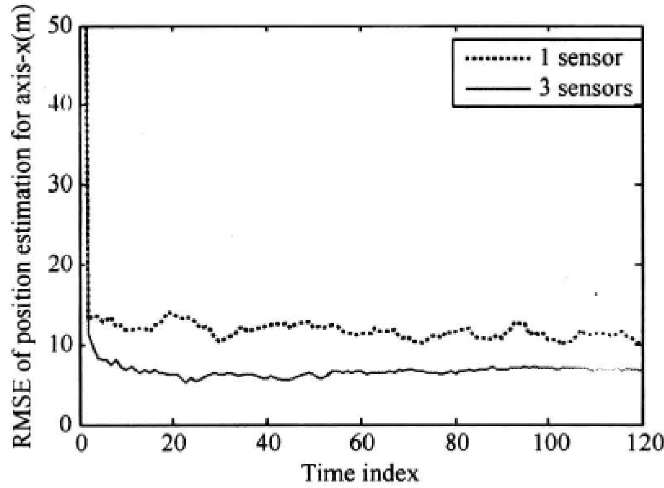


图 11.12 x 轴位置估计均方根误差

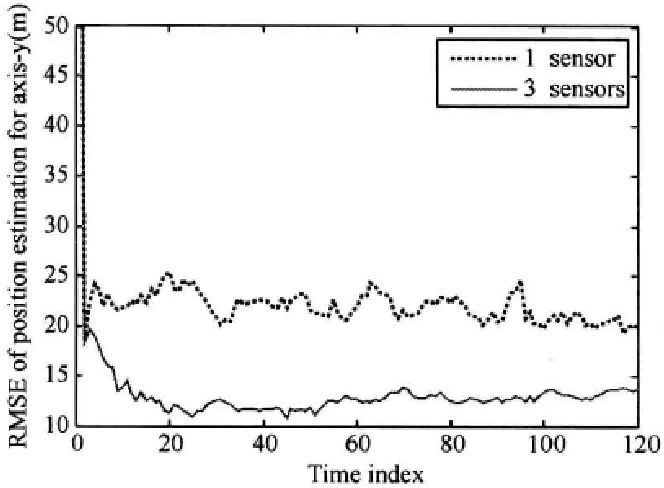


图 11.13 y 轴位置估计均方根误差

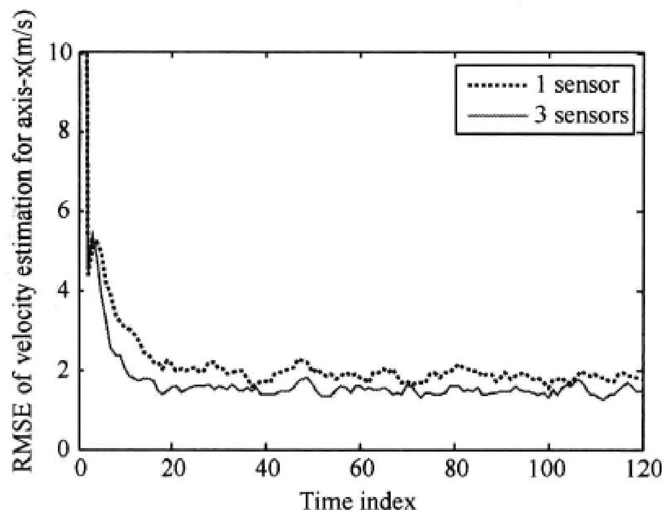


图 11.14 x 轴速度估计均方根误差

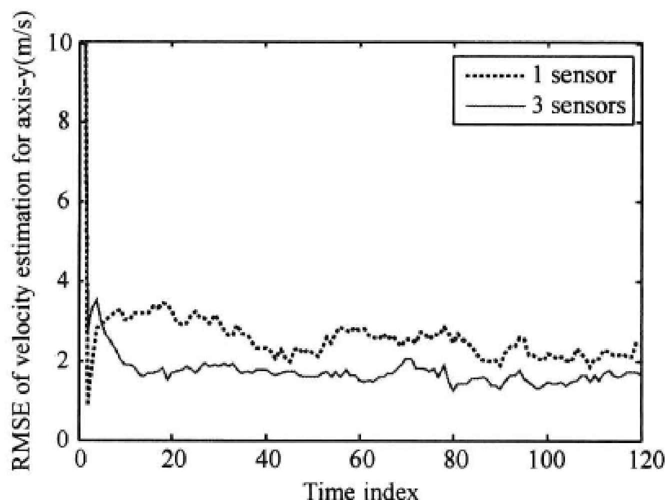


图 11.15 y 轴速度估计均方根误差

表 11.2 不同传感器配置下的跟踪结果比较

ARMSE / M	1 sensor					3 sensors				
	pos_x	pos_y	vel_x	vel_y	len	pos_x	pos_y	vel_x	vel_y	len
8000	11.02	20.87	1.81	2.40	0.70	6.54	12.59	1.48	1.57	0.51
6000	11.55	21.74	1.89	2.44	0.72	6.56	12.66	1.50	1.60	0.54
4000	13.10	23.92	1.99	2.62	0.79	7.73	15.36	1.55	1.65	0.68
2000	13.83	25.78	2.03	2.65	0.89	9.86	19.91	1.65	1.67	0.90

11.5 本章小结

本章主要阐述非线性贝叶斯滤波理论和方法在多传感器观测条件下的扩展，重点介绍了集中式和分布式这两种融合滤波原理，并针对集中式融合结构给出了基于解析高斯近似和随机采样近似方法的融合滤波实现过程和应用实例。

仿真结果表明，与单传感器相比，采用多传感器集中式融合的方法可以有效提高滤波精度和稳定性，特别是提高只测角条件下的目标跟踪能力。需要指出的是，对于分布式融合，由于每个节点传感器的滤波算法与基于传统的单传感器滤波算法相同，其主要差异主要体现在中心节点对各传感器估计结果（状态估计和协方差矩阵）的融合策略上。由于这部分内容不是本书分析的重点，本章没有进行深入展开，感兴趣的读者可参阅有关文献，如文献[13]和文献[14]等。

第 12 章　混合系统模型滤波

在前面讨论的各种非线性滤波算法中，主要基于单一的系统模型假设，即尽管系统状态是未知的，但其演化过程通常只有一种特定的形式。但在真实世界中，系统往往会因为有不同的输入而呈现出不同演化过程，相应地得到不同的量测，如在目标跟踪领域，出于规避或某种特定的需要等目的，往往会执行各种特定的机动方式，在战场上的军事目标更是如此。在这种条件下，需要寻求混合（Hybrid）系统滤波估计方法[15-17]来解决问题。

对混合系统进行估计通常采用多模（Multiple Model，MM）方法来处理，具体实现时需针对具体的混合模型形式进行。对于模式（Mode）固定的系统，即系统可能的演化模型个数固定不变，且不存在模式迁移（或转换）现象。在这种情况下，往往采用静态多模（Static Multiple Model）估计算法即可，即对每一种模式，采用相应的模型进行独立滤波估计，最后对各个基于模式的匹配滤波结果进行综合。这样，正确的模型必然是所考虑的模型当中的一个，且为模型后验概率最大的那一个。

实际中碰到的问题往往比上面所提到的要复杂，即系统模式存在转移或跳变（Jump）（如目标由匀速运动变为匀加速或匀转弯运动等），在这种条件下，只能采用动态多模（Dynamic Multiple Model）方法来处理。因此，下文如非特殊说明，所指的多模方法都默认为动态多模型方法。

多模方法利用多种状态模型来描述不同的系统状态演化模式，在系统状态矢量中增加一个离散变量用于指示该模型下的状态演化过程，实现系统状态与模型属性的混合估计。多模算法经历了三个发展阶段[18]，每一个阶段在结构、操作、性能，以及应用潜力和局限性方面都有着较大区别。第一代为自适应多模（AMM）方法；第二代为协作多模型（CMM）方法，最具有代表性的就是交互多模（IMM）方法；第三代为变结构多模型方法（VS-MM）。

从多模滤波的实现方法看，主要有基于混合高斯近似的方法和基于随机采样近似的方法，将书中第二部分介绍的混合高斯近似方法与 IMM 相结合是实现混合系统状态估计的常用处理手段，它在状态估计方面有着良好的性能。但是当采用随机采样近似方法与 IMM 的结合则存在一些问题：第一，粒子滤波本身是一种密度估计技术，而不是直接的状态估计方法，用于描述后验密度的粒子数量大，如果采用交互方式，算法复杂性较强；第二，粒子滤波算法是一种基于蒙特卡罗随机采样的近似技术，随机采样对于描述后验密度，搜索并逼近真实模式有着重要的意义，一旦采用交互处理，会破坏粒子状态转移的随机性。

鉴于此，本章主要考虑将混合高斯近似方法与 IMM 相结合（称为传统高斯混合交互多模，IMM）[19,20]，以及随机采样近似方法与 AMM 相结合（称为多模粒子滤波，MMPF）[21,22]的混合模型滤波算法，并针对机动目标跟踪应用问题给出了具体的实例。

12.1 混合状态的贝叶斯估计

离散时间混合系统可由如下动态方程和观测方程描述

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}, r_k, \boldsymbol{w}_{k-1}) \tag{12-1}$$

$$\boldsymbol{z}_k = h(\boldsymbol{x}_k, r_k, \boldsymbol{v}_k) \tag{12-2}$$

式中：r_k 为 $k-1$ 到 k 时刻之间起作用的模型变量，该变量通常描述为时间均匀的一阶马尔可夫链，其转移概率为

$$\pi_{ij} \triangleq P\{r_k = j \mid r_{k-1} = i\} \qquad (i, j \in \mathbb{M}) \tag{12-3}$$

式中：$\mathbb{M} \triangleq \{1, 2, \cdots, m\}$。转移概率矩阵 $\boldsymbol{\Pi} = [\pi_{ij}]$ 为 $m \times m$ 方阵，且矩阵元素满足

$$\pi_{ij} \geqslant 0, \quad \sum_{j=1}^{m} \pi_{ij} = 1 \tag{12-4}$$

对于每一个 $i, j \in \mathbb{M}$，初始模型概率表示为

$$\mu_i \triangleq P\{r_0 = i\} \tag{12-5}$$

而对于每一个 μ_i，满足

$$\mu_i \geqslant 0, \quad \sum_{i=1}^{m} \mu_i = 1 \tag{12-6}$$

式（12-1）和式（12-2）的其他部分在前文中已有过描述。当 $m=1$ 时，式（12-1）和式（12-2）分别等同于式（1.1）和式（1.2）。而由式（12-1）和式（12-2）定义的非线性滤波问题，不仅要估计系统状态（连续变量），还要估计模型值（离散变量），称这种类型的问题为混合状态估计问题。

在贝叶斯滤波框架内，式（12-1）和式（12-2）的概念性解也与式（1.3）和式（1.4）类似，其区别在于现在所考虑的状态矢量为混合状态矢量 $\boldsymbol{y}_k = [\boldsymbol{x}_k^{\mathrm{T}}, r_k]^{\mathrm{T}}$。此时，根据贝叶斯定理，其递推处理过程可按照预测和更新两个步骤描述如下。

预测：

$$p\left(\boldsymbol{x}_k, r_k = j \mid \boldsymbol{z}_{1:k-1}\right) = \sum_i \pi_{ij} \int p\left(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}, r_k = j\right) p\left(\boldsymbol{x}_{k-1}, r_{k-1} = i \mid \boldsymbol{z}_{1:k-1}\right) \mathrm{d}\boldsymbol{x}_{k-1} \tag{12-7}$$

更新：

$$p\left(\boldsymbol{x}_k, r_k = j \mid \boldsymbol{z}_{1:k}\right) = \frac{p\left(\boldsymbol{z}_k \mid \boldsymbol{x}_k, r_k = j\right) p\left(\boldsymbol{x}_k, r_k = j \mid \boldsymbol{z}_{1:k-1}\right)}{\sum_j \int p\left(\boldsymbol{z}_k \mid \boldsymbol{x}_k, r_k = j\right) p\left(\boldsymbol{x}_k, r_k = j \mid \boldsymbol{z}_{1:k-1}\right) \mathrm{d}\boldsymbol{x}_k} \tag{12-8}$$

上述得到的是混合系统模型估计的一般性求解表达式，具体的求解方法依赖于具体的模型形式。鉴于混合高斯滤波和粒子滤波在处理非线性/非高斯估计问题中的强大能力，接下来首先考虑基于混合高斯近似的多模型滤波算法，在此基础上分析基于粒子实现的多模滤波算法。

12.2 多模高斯滤波算法

考虑有限个（m 个）模型集条件下的混合估计问题，根据全概公式，系统状态的后验 PDF 为

$$\begin{aligned}p\left(\boldsymbol{x}_k \mid z_{1:k}\right)&=\sum_{j=1}^{m} p(\boldsymbol{x}_k \mid r_k(j),z_{1:k})\Pr(r_k(j)|z_{1:k})\\&=\sum_{j=1}^{m} p(\boldsymbol{x}_k \mid r_k(j),z_{1:k-1},z_k)\mu_k(j)\end{aligned} \tag{12-9}$$

以各模型为条件的状态后验 PDF 为

$$p(\boldsymbol{x}_k \mid r_k(j),z_{1:k-1},z_k)=\frac{p(z_k \mid r_k(j),\boldsymbol{x}_k)}{p(z_k \mid r_k(j),z_{1:k-1})}p(\boldsymbol{x}_k \mid r_k(j),z_{1:k-1}) \tag{12-10}$$

将全概公式再次用于式（12-10）中的第二项，有

$$p(\boldsymbol{x}_k \mid r_k(j),z_{1:k-1})=\sum_{i=1}^{m} p(\boldsymbol{x}_k \mid r_k(j),r_{k-1}(i),z_{1:k-1})\Pr[r_{k-1}(i)|r_k(j),z_{1:k-1}] \tag{12-11}$$

在高斯近似假设条件下，式（12-11）可进一步表示为

$$\begin{aligned}p(\boldsymbol{x}_k \mid r_k(j),z_{1:k-1})&\approx\sum_{i=1}^{m} p[\boldsymbol{x}_k \mid r_k(j),r_{k-1}(i),\{\hat{\boldsymbol{x}}_{k-1}^{l},\boldsymbol{P}_{k-1}^{l}\}_{l=1}^{m}]\mu_{k-1}[i|j]\\&=\sum_{i=1}^{m} p[\boldsymbol{x}_k \mid r_k(j),r_{k-1}(i),\hat{\boldsymbol{x}}_{k-1}^{i},\boldsymbol{P}_{k-1}^{i}]\mu_{k-1}[i|j]\end{aligned} \tag{12-12}$$

式（12-12）第一行表示前 $k-1$ 个时刻的状态可近似为 m 个基于模型的条件状态估计之和。

进一步假定该混合形式为高斯混合分布，而后对单个高斯分量采用矩匹配近似处理，得

$$\begin{aligned}p(\boldsymbol{x}_k \mid r_k(j),z_{1:k-1})&\approx\sum_{i=1}^{m}\mathcal{N}\left(\boldsymbol{x}_k;E[\boldsymbol{x}_k|r_k(j),\hat{\boldsymbol{x}}_{k-1}^{i}],\mathrm{cov}(\cdot)\right)\mu_{k-1}[i|j]\\&\approx\mathcal{N}\left(\boldsymbol{x}_k;\sum_{i=1}^{m}E[\boldsymbol{x}_k|r_k(j),\hat{\boldsymbol{x}}_{k-1}^{i}]\mu_{k-1}[i|j],\mathrm{cov}(\cdot)\right)\\&=\mathcal{N}\left(\boldsymbol{x}_k;E\left[\boldsymbol{x}_k|r_k(j),\sum_{i=1}^{m}\hat{\boldsymbol{x}}_{k-1}^{i}\mu_{k-1}[i|j]\right],\mathrm{cov}(\cdot)\right)\end{aligned} \tag{12-13}$$

式（12-13）最后一行利用了混合高斯分布的线性性质，与模型 j 匹配的滤波器的输入是由 m 个滤波器的交互得到的，每个滤波的混合分量为 $\hat{\boldsymbol{x}}_{k-1}^{i}$，相应的权重（或概率）为 $\mu_{k-1}[i|j]$，称为混合概率。

利用式（12-13）实现的多模滤波算法称为交互多模（IMM）算法，图 12.1 给出了该算法的实现结构，包含 m 个并行运行的交互滤波器，混合过程在每个滤波器的输入阶段完成，并以 $z_{1:k-1}$ 为条件，混合概率的计算将在下文的具体算法流程中详细说明。

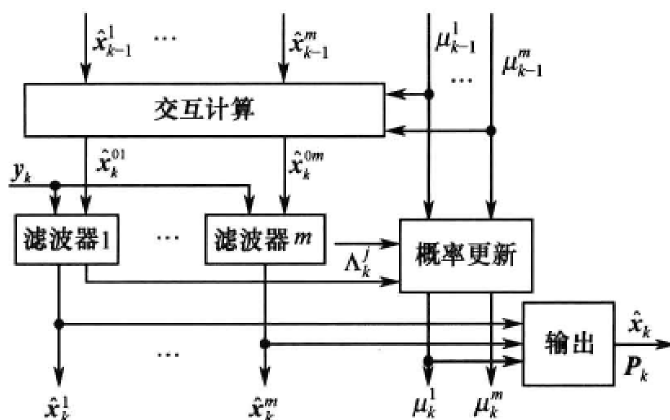


图 12.1 IMM 估计算法

交互多模滤波的详细算法流程可概括如下。

Alg（12.1）：Interacting Multiple Model Filtering

步骤 1：混合概率计算

在 $z_{1:k-1}$ 和 k 时刻模式 r_j 已知的条件下，$k-1$ 时刻模式 r_i 的概率为

$$\begin{aligned}\mu_{k-1}[i|j] &\triangleq \Pr[r_{k-1}(i)|r_k(j),z_{1:k-1}] \\ &= \frac{1}{\bar{c}_j}\Pr[r_k(j)|r_{k-1}(i),z_{1:k-1}]\Pr[r_{k-1}(i)|z_{1:k-1}]\end{aligned} \tag{12-14}$$

式（12-14）的混合概率可进一步表示为

$$\mu_{k-1}[i|j]=\frac{1}{\bar{c}(j)}\pi_{i,j}\mu_{k-1}(i)\ ,\ \ i,j=1,2,\cdots,m \tag{12-15}$$

式中：归一化常数 $\bar{c}(j)=\sum_{i=1}^{m}\pi_{i,j}\mu_{k-1}(i)$。

步骤 2：模型混合

由 $\hat{\boldsymbol{x}}_{k-1}^{i}$ 出发，根据式（12-13）计算与 $r_k(j)$ 相对应的混合初始条件，即

$$\hat{\boldsymbol{x}}_{k-1}(0j)=\sum_{i=1}^{m}\hat{\boldsymbol{x}}_{k-1}^{i}\mu_{k-1}[i|j]\ ,\ \ j=1,2,\cdots,m \tag{12-16}$$

相应的协方差为

$$\boldsymbol{P}_{k-1}(0j)=\sum_{i=1}^{m}\mu_{k-1}[i|j]\{(\hat{\boldsymbol{x}}_{k-1}^{i}-\hat{\boldsymbol{x}}_{k-1}^{0j})(\hat{\boldsymbol{x}}_{k-1}^{i}-\hat{\boldsymbol{x}}_{k-1}^{0j})^{\mathrm{T}}+\boldsymbol{P}_{k-1}^{i}\}\ ,\ \ j=1,2,\cdots,m \tag{12-17}$$

步骤 3：模型匹配滤波

利用步骤 2 的结果作为模式 $r_k(j)$ 的匹配滤波的输入，在量测 z_k 的作用下得到 $\hat{\boldsymbol{x}}_k^j$ 和 $\boldsymbol{P}_k^j$，对应于 m 个滤波器的似然函数为

$$\begin{aligned}\Lambda_k(j)&=p[z_k|r_k(j),z_{1:k-1}]=p[z_k|r_k(j),\boldsymbol{x}_{k-1}(0j),\boldsymbol{P}_{k-1}(0j)] \\ &\approx \mathcal{N}[z_k;\hat{z}_{k|k-1}(j),\boldsymbol{S}_k(j)]\end{aligned} \tag{12-18}$$

步骤 4：模式概率更新

$$
\begin{aligned}
\mu_k(j) &\triangleq \Pr[r_k(j)|z_{1:k}] = \frac{1}{c} p[z_k|r_k(j), z_{1:k-1}]\Pr[r_k(j)|z_{1:k-1}] \\
&= \frac{1}{c}\Lambda_k(j)\sum_{i=1}^{m} p[r_k(j)|r_{k-1}(i), z_{1:k-1}]\Pr[r_{k-1}(i)|z_{1:k-1}] \\
&= \frac{1}{c}\Lambda_k(j)\sum_{i=1}^{m}\pi_{i,j}\mu_{k-1}(i)
\end{aligned}
\tag{12-19}
$$

式中：$c=\sum_{j=1}^{m}\Lambda_k(j)\bar{c}(j)$。

步骤 5：状态和协方差估计

基于模型的条件估计及协方差为

$$
\hat{\boldsymbol{x}}_k = \sum_{j=1}^{m}\hat{\boldsymbol{x}}_k(j)\mu_k(j) \tag{12-20}
$$

$$
\boldsymbol{P}_k = \sum_{j=1}^{m}\mu_k[j]\{(\hat{\boldsymbol{x}}_k^j - \hat{\boldsymbol{x}}_k)(\hat{\boldsymbol{x}}_k^j - \hat{\boldsymbol{x}}_k)^{\mathrm{T}} + \boldsymbol{P}_k^j\},\ j=1,2,\cdots,m \tag{12-21}
$$

需要指出的是，步骤 5 只用于结果输出，并不参与递推计算。另外，式（12-18）中包含了量测的预测 $\hat{z}_{k|k-1}(j)$ 及预测误差协方差矩阵 $\boldsymbol{S}_k(j)$，这两式的计算依赖于具体的观测模型和非线性滤波算法，典型的有 EKF、UKF 或 QKF 等，相应地得到基于 EKF/UKF/QKF 的 IMM 估计算法。

12.3 多模粒子滤波算法

对式（12-1）和式（12-2）求解也可采用基于 SMC 近似的多模粒子滤波来实现。定义新的状态向量为 $\boldsymbol{y}_k = [\boldsymbol{x}_k^{\mathrm{T}}, r_k]^{\mathrm{T}}$，其中 $r_k \in \mathbb{M} = \{1,2,\cdots,m\}$。且假设已知初始密度 $p(\boldsymbol{x}_0)$ 和 $p(r_1) = \sum_{i=1}^{m}\mu_i\delta(r_1 - i)$。另外，令 $\{\boldsymbol{y}_k^{(n)}, w_k^{(n)}\}_{n=1}^{N}$ 为描述后验分布 $p(\boldsymbol{y}_k \mid z_{1:k})$ 的粒子集。则多模粒子滤波算法可概括如下。

Alg（12.2）：Multiple Model Particle Filtering

$\left[\{\boldsymbol{y}_k^{(n)}, w_k^{(n)}\}_{n=1}^{N}\right] = \mathrm{MMPF}\left[\{\boldsymbol{y}_{k-1}^{(n)}, w_{k-1}^{(n)}\}_{n=1}^{N}, \boldsymbol{z}_k\right]$

•按照 Alg（12.3）进行模型状态转移

$\left[\{r_k^{(n)}\}_{n=1}^{N}\right] = \mathrm{MT}\left[\{r_{k-1}^{(n)}\}_{n=1}^{N}, \boldsymbol{\Pi}\right]$

•按照 Alg（12.4）进行模型的 SIS 滤波

$\left[\{\boldsymbol{x}_k^{(n)}, w_k^{(n)}\}_{n=1}^{N}\right] = \text{M-SIS}\left[\{\boldsymbol{x}_{k-1}^{(n)}, r_{k-1}^{(n)}, w_{k-1}^{(n)}\}_{n=1}^{N}, \boldsymbol{z}_k\right]$

•计算有效样本数 $\hat{N}_{eff}$

•if $\hat{N}_{eff} < N_s$

$\left[\{\boldsymbol{y}_k^{(n)}, N^{-1}\}_{n=1}^{N}\right] = \mathrm{resample}\left[\{\boldsymbol{y}_k^{(n)}, w_k^{(n)}\}_{n=1}^{N}\right]$

•end if

MMPF 算法的第一步是使用时刻 $k-1$ 的模型粒子集 $\{r_{k-1}^{(n)}\}_{n=1}^{N}$ 和转移概率矩阵 $\Pi=\{\pi_{ij}\}$ 产生时刻 k 的模型粒子集 $\{r_k^{(n)}\}_{n=1}^{N}$，其实现过程如 Alg（12.3）描述的模型变量状态转移（Model Transition，MT）算法。

Alg（12.3）：Model Transition

$$\left[\{r_k^{(n)}\}_{n=1}^{N}\right]=\text{MT}\left[\{r_{k-1}^{(n)}\}_{n=1}^{N},\Pi\right]$$

- for $i=1:m$
  - $c_i(0)=0$
  - for $j=1:m$
    - * $c_i(j)=c_i(j-1)+\pi_{ij}$
  - end for
- end for
- for $n=1:N$
  - 产生均匀分布数 $u_n\sim\mathcal{U}[0,1]$
  - 令 $i=r_{k-1}^{(n)}$
  - $m_0=1$
  - while $\left(c_i(m_0)<u_n\right)$
    - * $m_0=m_0+1$
  - end while
  - 令 $r_k^{(n)}=m_0$
- end for

MT 算法的基本处理步骤是：如果 $r_{k-1}^{(n)}=i$，$u_n\sim\mathcal{U}[0,1]$，当下式成立时，$r_k^{(n)}$ 的取值为 $m_0\in\mathbb{M}$。

$$\sum_{j=1}^{m_0-1}\pi_{ij}<u_n\leqslant\sum_{j=1}^{m_0}\pi_{ij} \tag{12-22}$$

式中：$\sum_{j=1}^{m_0}\pi_{ij}$ 表示离散随机变量 r_k 在给定 $r_{k-1}=i$ 条件下的累积分布函数。

多模粒子滤波算法的第二步是进行基于模型的序贯重要采样滤波（Model Based SIS filter，M-SIS），具体实现过程可总结如下。

Alg（12.4）：Model Based SIS Filtering

$$\left[\{\boldsymbol{x}_k^{(n)},w_k^{(n)}\}_{n=1}^{N}\right]=\text{M-SIS}\left[\{\boldsymbol{x}_{k-1}^{(n)},r_{k-1}^{(n)},w_{k-1}^{(n)}\}_{n=1}^{N},\boldsymbol{z}_k\right]$$

- for $n=1:N$
  - 粒子采用：$\boldsymbol{x}_k^{(n)}\sim q(\boldsymbol{x}_k\mid\boldsymbol{x}_{k-1}^{(n)},r_{k-1}^{(n)},\boldsymbol{z}_k)$
  - 计算重要密度权重

$$w_k^{(n)}=w_{k-1}^{(n)}\frac{p(\boldsymbol{z}_k\mid\boldsymbol{x}_k^{(n)},r_k^{(n)})p(\boldsymbol{x}_k^{(n)}\mid\boldsymbol{x}_{k-1}^{(n)},r_k^{(n)})}{q(\boldsymbol{x}_k\mid\boldsymbol{x}_{k-1}^{(n)},r_{k-1}^{(n)},\boldsymbol{z}_k)}$$

- end for

$$\bullet \text{计算权重和}\ w_{sum}=\sum_{n=1}^{N}w_k^{(n)}$$

$$\bullet \text{for}\ n=1:N$$

$$-\text{权重归一化：}\ \tilde{w}_k^{(n)}=w_k^{(n)}/w_{sum}$$

$$\bullet \text{end for}$$

Alg（12.4）中的最优重要度密度函数为

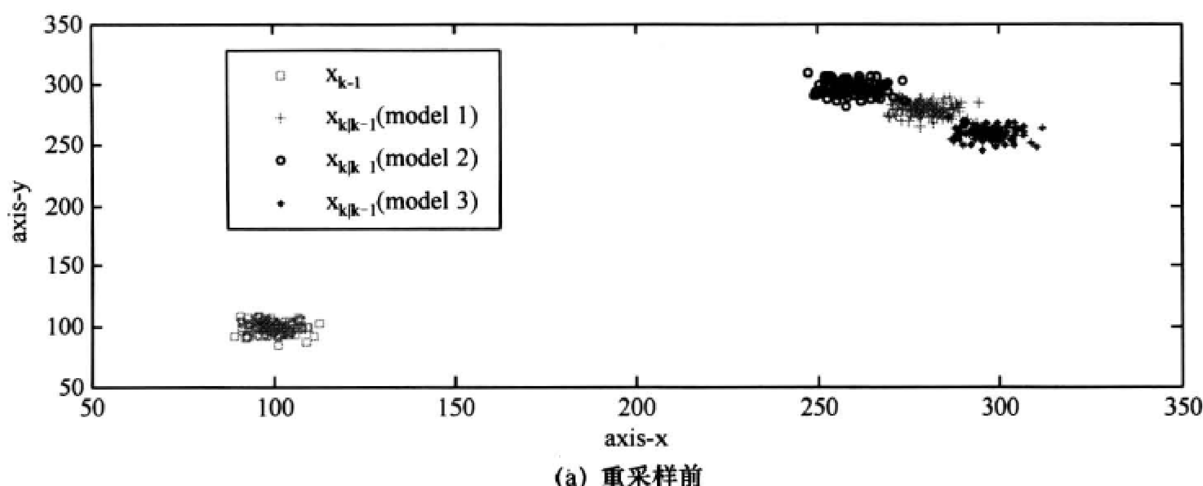
$$q(\boldsymbol{x}_k\mid \boldsymbol{x}_{k-1}^{(n)},r_{k-1}^{(n)},\boldsymbol{z}_k)_{opt}=p(\boldsymbol{x}_k\mid \boldsymbol{x}_{k-1}^{(n)},r_{k-1}^{(n)},\boldsymbol{z}_k) \tag{12-23}$$

而通常将其选为先验状态转移密度函数

$$q(\boldsymbol{x}_k\mid \boldsymbol{x}_{k-1}^{(n)},r_{k-1}^{(n)},\boldsymbol{z}_k)=p(\boldsymbol{x}_k\mid \boldsymbol{x}_{k-1}^{(n)},r_{k-1}^{(n)}) \tag{12-24}$$

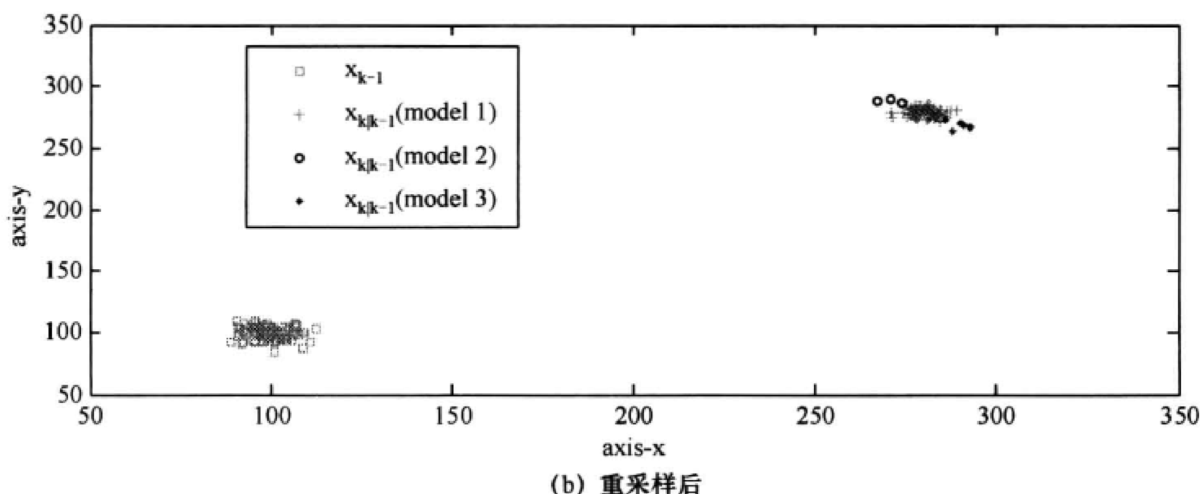
在 MMPF 算法中，一个重要的现实问题是，如果各个粒子按照各自选择的模型进行状态预测，粒子滤波是否能够保证k时刻的粒子集能够准确描述该时刻的目标状态分布。众所周知，粒子权重大小表示了粒子状态接近真实状态的程度。若用 MMPF 算法解决混合系统的状态估计问题，在粒子状态中增加了表示状态模型的变量，滤波过程中，先通过$k-1$时刻的粒子模型变量值和模型转移概率来预测k时刻的粒子模型变量值。然后，$k-1$时刻的各个粒子依据新模型变量值对应的状态转移方程进行粒子状态预测，得到k时刻的粒子，并计算粒子权重。此时，各个粒子权重大小不一，符合目标真实运动模型的粒子状态将接近于真实状态，其权值相对较大；反之，则远离目标真实状态，权重较小。在这种情况下，可以发挥重采样方法所固有的作用，重复权重大的粒子，剔除掉权重小的粒子，这样，将会有大多数粒子的模型变量值等于真实值，粒子的分布更接近目标真实状态分布。下面通过一个简单的示例，说明多模条件下的粒子预测及重采样过程对模型的选择能力。

如图 12.2 所示，假定$k-1$时刻目标的真实状态为$\boldsymbol{x}_{k-1}$=[100 60 100 60]^{T}，用 100 个散布在真值位置附近的粒子近似目标的状态分布，散布方差为$\boldsymbol{P}_{k-1}$=diag([5^2 1^2 5^2 1^2])，分别用匀速直线模型、顺时针匀转弯模型及逆时针匀转弯模型进行预测（采样间隔为 3s，转弯率为 4°/s），得到的粒子云分布情况如图（a）所示。进一步假设在$k-1$到k时刻目标按照模型 1 运动，k时刻目标的真实状态为$\boldsymbol{x}_{k-1}$=[280 60 280 60]^{T}，而此时得到的目标位置观测为(281.54, 277.75)，观测误差协方差为$\boldsymbol{R}$=diag([25 25])。图 12.2（b）为根据k时刻的观测进行权重计算，重采样之后的粒子分布。



(a) 重采样前

图 12.2　多模粒子的预测与更新过程



(b) 重采样后

图 12.2　多模粒子的预测与更新过程（续）

从图 12.2 中的结果可以清楚地看出，多模可以充分挖掘目标的运动状态变化，而粒子滤波算法通过重采样步骤能够有效选择模型变量，使其逼近真实的目标运动状态。

12.4　应 用 实 例

12.4.1　交互多模机动目标跟踪

假设二维平面中的目标运动方程满足

$$\boldsymbol{x}_k = \boldsymbol{F}\boldsymbol{x}_{k-1} + \boldsymbol{\Gamma}\boldsymbol{u}_{k-1} + \boldsymbol{G}\boldsymbol{w}_{k-1} \tag{12-25}$$

式中：$\boldsymbol{x}_k=[x_k \ y_k \ \dot{x}_k \ \dot{y}_k]^{\mathrm{T}}$ 为 k 时刻目标的状态；$\boldsymbol{u}_k=[a_x \ a_y]^{\mathrm{T}}$ 为控制项；$\boldsymbol{F}$ 状态转移矩阵；$\boldsymbol{\Gamma}$ 为控制输入矩阵；$\boldsymbol{G}$ 为噪声输入矩阵；$\boldsymbol{w}_{k-1}$ 为零均值高斯过程噪声；且

$$\boldsymbol{F}=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & T & 0\\ 0 & 0 & 0 & T\end{bmatrix},\quad \boldsymbol{\Gamma}=\boldsymbol{G}=\begin{bmatrix}T^2/2 & 0\\ T^2/2 & 0\\ 0 & T\\ 0 & T\end{bmatrix},\quad \mathrm{cov}[\boldsymbol{w}_k]=\boldsymbol{Q}_k$$

采用主动雷达传感器，观测量包括目标距离 r_k、距离变化率 $\dot{r}_k$ 及方位 φ_k，观测方程可表示为

$$\boldsymbol{y}_k=h(\boldsymbol{x}_k)+\boldsymbol{v}_k=\begin{bmatrix}\sqrt{x_k^2+y_k^2}\\ \dfrac{x_k\dot{x}_k+y_k\dot{y}_k}{\sqrt{x_k^2+y_k^2}}\\ \arctan(y_k/x_k)\end{bmatrix}+\begin{bmatrix}v_{r,k}\\ v_{\dot{r},k}\\ v_{\varphi,k}\end{bmatrix} \tag{12-26}$$

式中：$\boldsymbol{v}_k$ 为零均值高斯噪声，且 $\mathrm{cov}[\boldsymbol{v}_k]=\boldsymbol{R}_k$。

假设目标的初始状态为 $\boldsymbol{x}_0=[80\mathrm{km} \ \ 20\mathrm{km} \ \ -200\mathrm{m/s} \ \ 0\mathrm{m/s}]^{\mathrm{T}}$，过程噪声协方差矩阵 $\boldsymbol{Q}_k=\mathrm{diag}([(1\mathrm{m/s^2})^2 \ (1\mathrm{m/s^2})^2])$，采样间隔 T=1s。目标的航迹分为 5 段，每一段的运动状态分别描述为：

第 1 段（0s～40s）：以 $\boldsymbol{x}_0$ 为初始状态的匀速运动。

第 2 段（41s～90s）：第一次机动，加速度 $\boldsymbol{u}$=[0.6g 0.8g]^{T}（g 为重力加速度）。

第 3 段（91s～140s）：以第一次机动后为初始状态的匀速运动。

第 4 段（141s～180s）：第二次机动，加速度 $\boldsymbol{u}$=[－0.6g －0.8g]^{T}。

第 5 段（181s～240s）：以第二次机动后为初始状态的匀速运动。

采用包含 CV 和 CV 两种模型的交互多模（IMM）算法进行跟踪，具体应用时，由于 CV 模型的状态量维数为 4，为此需要对状态量进行扩展，得到用于跟踪的 CV 模型为

$$\boldsymbol{x}_k=\boldsymbol{F}_{\mathrm{CV}}\boldsymbol{x}_{k-1}+\boldsymbol{\Gamma w}_k=\begin{bmatrix}1&0&T&0&0&0\\0&1&0&T&0&0\\0&0&1&0&0&0\\0&0&0&1&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\end{bmatrix}\boldsymbol{x}_{k-1}+\begin{bmatrix}T^2/2&0\\0&T^2/2\\T&0\\0&T\\0&0\\0&0\end{bmatrix}\boldsymbol{w}_k$$

式中

$$\boldsymbol{x}_k=[x\ y\ \dot{x}\ \dot{y}\ \ddot{x}\ \ddot{y}]^{\mathrm{T}}(k)$$

同理，得到用于跟踪的 CA 模型为

$$\boldsymbol{x}_k=\boldsymbol{F}_{\mathrm{CA}}\boldsymbol{x}_{k-1}+\boldsymbol{\Gamma w}_k=\begin{bmatrix}1&0&T&0&T^2/2&0\\0&1&0&T&0&T^2/2\\0&0&1&0&T&0\\0&0&0&1&0&T\\0&0&0&0&1&0\\0&0&0&0&0&1\end{bmatrix}\boldsymbol{x}_{k-1}+\begin{bmatrix}T^2/2&0\\0&T^2/2\\T&0\\0&T\\1&0\\0&1\end{bmatrix}\boldsymbol{w}_k$$

假定初始状态估计误差协方差矩阵为 $\boldsymbol{P}_0^1=\boldsymbol{P}_0^2=\mathrm{diag}([300^2\ 300^2\ 100^2\ 100^2\ 2^2\ 2^2])$，初始模型概率 $[\mu_0(1)\ \mu_0(2)]=[0.95\ 0.05]$，模型转移概率矩阵 $\pi=\begin{bmatrix}0.95&0.05\\0.10&0.90\end{bmatrix}$，距离、方位及径向速度观测精度分别为 $\sigma_r=5\mathrm{m}$、$\sigma_\varphi=0.2^\circ$ 和 $\sigma_{\dot{r}}=0.2\mathrm{m/s}$。采用随机初始化方法产生初始状态估计，利用第 4 章中介绍的 UKF 算法进行跟踪滤波，实验次数为 500，所得结果如图 12.3～图 12.6 所示。

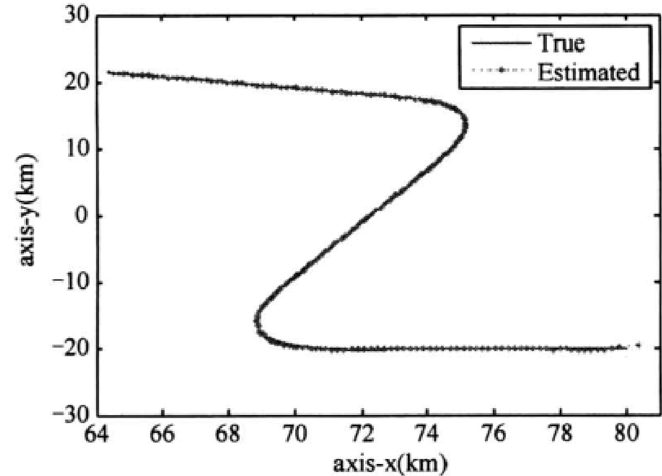


图 12.3　单次跟踪示例

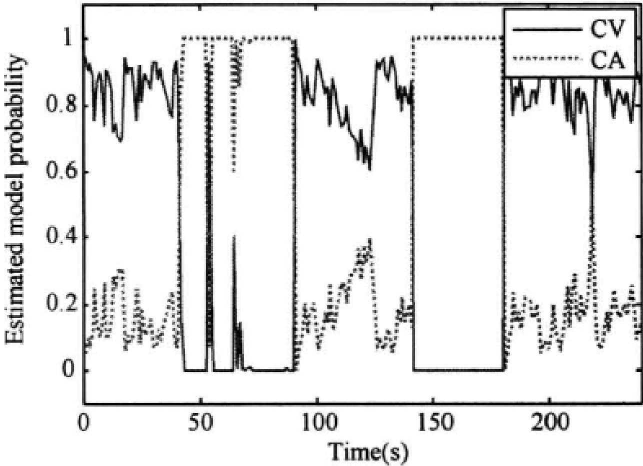


图 12.4　模型概率估计

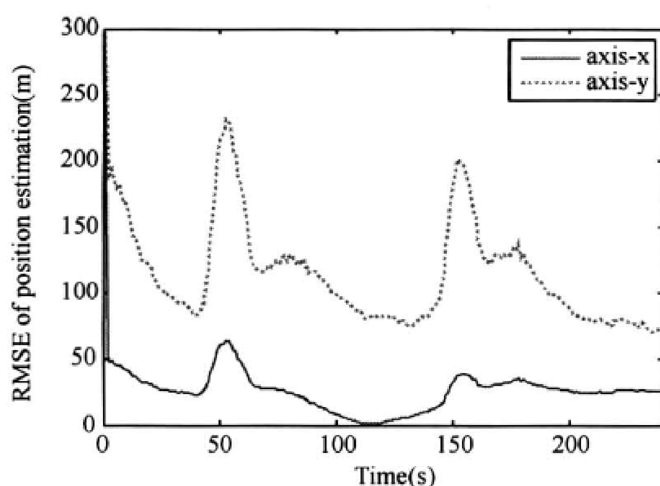


图 12.5　位置估计均方根误差

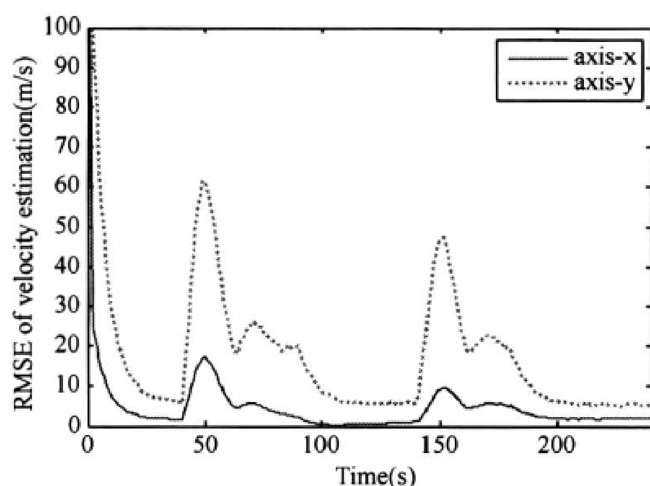


图 12.6　速度估计均方根误差

由图 12.3～图 12.6 中的结果可以看出，采用 IMM 算法可以对机动目标进行有效跟踪，不过从图中可以看出，在机动发生时刻跟踪误差有短时上升的现象。这是因为跟踪过程隐含了对未知加速度的估计，这个估计从初始建立到稳定收敛有个过程，从而使估计误差出现先增后降的趋势。另外，从图 12.4 中的结果可以看到，在目标处于近匀速阶段，匀加速运动模型的模型概率并不接近零，这是因为匀加速运动模型包含了目标的匀速运动，因而具备近似匀速运动的能力，这种现象即所谓的存在模型竞争现象。只不过在这种条件下，CA 模型是失配的，因而模型概率不如 CV 模型高。

12.4.2　多模跟踪的粒子实现

考虑二维平面内的目标检测前跟踪问题，假设在观测时间内目标的运动满足以下种模型中的一种：①匀速运动；②顺时针转弯；③逆时针转弯，其运动状态可用通式表示为

$$\boldsymbol{x}_{k+1}=\boldsymbol{F}^{(r_k)}\boldsymbol{x}_k+\boldsymbol{w}_k \tag{12-27}$$

式中：$\boldsymbol{x}_k=[x_k\ \dot{x}_k\ y_k\ \dot{y}_k\ I_k]^{\mathrm{T}}$ 为状态矢量；$\boldsymbol{w}_k$ 为过程噪声且 $\boldsymbol{w}_k\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{Q}_k)$；$r_k\in\mathbb{M}=\{1,2,3\}$ 为 $k-1$ 到 k 时刻之间的模型状态变量；$\boldsymbol{F}^{(r_k)}$ 为相应的状态转移矩阵，针对以上 3 种模型，其转换矩阵的具体形式为

$$\boldsymbol{F}^{(1)}=\begin{bmatrix}1 & T & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 1 & T & 0\\ 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 1\end{bmatrix}$$

$$\boldsymbol{F}^{(s)}=\begin{bmatrix}1 & \sin(\omega_k^{(s)}T)/\omega_k^{(s)} & -(1-\cos(\omega_k^{(s)}T))/\omega_k^{(s)} & 0 & 0\\ 0 & \cos(\omega_k^{(s)}T) & 0 & \sin(\omega_k^{(s)}T) & 0\\ 0 & (1-\cos(\omega_k^{(s)}T))/\omega_k^{(s)} & 1 & \sin(\omega_k^{(s)}T)/\omega_k^{(s)} & 0\\ 0 & \sin(\omega_k^{(s)}T) & 0 & \cos(\omega_k^{(s)}T) & 0\\ 0 & 0 & 0 & 0 & 1\end{bmatrix},\quad s=2,3$$

式中：T 为传感器扫描周期；$\omega_k^{(2)}=\dfrac{-\alpha_m}{\sqrt{\dot{x}_k^2+\dot{y}_k^2}}$；$\omega_k^{(3)}=\dfrac{\alpha_m}{\sqrt{\dot{x}_k^2+\dot{y}_k^2}}$；$\alpha_m$ 为机动加速度。

在某一特定时刻，目标只遵循一种运动模型；但在整个运动过程中，目标运动可在各种模型之间相互转换，其转换规律可用马尔可夫转移概率矩阵来描述，该矩阵可定义为

$$\Pi_m=\begin{bmatrix}\mu_{11} & \mu_{12} & \mu_{13}\\ \mu_{21} & \mu_{22} & \mu_{23}\\ \mu_{31} & \mu_{32} & \mu_{33}\end{bmatrix}$$

式中：$\mu_{ij}(i,j=1,2,3)$ 为模型 i 到模型 j 的转移概率。

k 时刻目标可能存在于传感器警戒区域内，也可能不存在，可用变量 $E_k\in\{0,1\}$ 进行建模。其中 0 表示目标不存在，1 表示目标存在。此外，新生目标和消亡目标的概率分别为

$$P_b\triangleq P\{E_k=1|E_{k-1}=0\}$$
$$P_d\triangleq P\{E_k=0|E_{k-1}=1\}$$

相应地，转移概率矩阵为

$$\Pi_e=\begin{bmatrix}1-P_b & P_b\\ P_d & 1-P_d\end{bmatrix}$$

设在一个大小为 $n_x\times n_y$ 的警戒区域内，传感器的观测方程为

$$z_k^{(i,j)}=\begin{cases}h_k^{(i,j)}(\boldsymbol{x}_k)+v_k^{(i,j)} & E_k^{(i,j)}=1\\ v_k^{(i,j)} & E_k^{(i,j)}=0\end{cases} \tag{12-28}$$

式中：$h_k^{(i,j)}(\boldsymbol{x}_k)=\dfrac{\Delta_x\Delta_y I_k}{2\pi\Sigma^2}\exp\left[-\dfrac{(i\Delta_x-x_k)^2+(j\Delta_y-y_k)^2}{2\Sigma^2}\right]$，$I_k$ 为信号强度，Σ 为扩散因子；$v_k^{(i,j)}$ 为第 (i,j) 个分辨单元的观测噪声。

下面结合上述系统模型具体分析多模粒子算法的实现过程，为此首先引入状态变量 $\boldsymbol{y}_k=[\boldsymbol{x}_k^{\mathrm{T}},r_k]^{\mathrm{T}}$。假设已知在 $k-1$ 时刻的后验概率密度 $p(\boldsymbol{y}_{k-1}|z_{1:k-1})$ 可由粒子集 $\{\boldsymbol{y}_{k-1}^{(n)},w_{k-1}^{(n)}\}_{n=1}^{N}$ 来表征（N 为粒子数目），则其完整的一步递推过程可概括如下。

Alg（12.5）：MMPF Based Track-before-Detect

步骤 1：存在变量预测

根据 $E_{k-1}^{(n)}$ 和转移概率矩阵 Π_e，得到 k 时刻存在变量 $E_k^{(n)}(n=1,2,\cdots,N)$。

步骤 2：粒子状态预测

这里仅对 $E_k^{(n)}=1$ 的粒子进行预测，分两种情况分别进行处理。

（1）对“新生”粒子（即满足 $E_{k-1}^{(n)}=0$，$E_k^{(n)}=1$ 的粒子），对于位置变量 $(x_k^{(n)},y_k^{(n)})$ 选择强度超过预设门限 γ 的网格单元集（$z_k^{(i,j)}>\gamma$），使粒子在这些单元内随机分布；对于速度变量 $(\dot{x}_k^{(n)}\ \dot{y}_k^{(n)})$，则使其在 $[-v_{\max}\ v_{\max}]$ 之间均匀分布；同样强度分布满足 $I_k^{(n)}\sim\mathcal{U}[I_{\min}\ I_{\max}]$；同时根据初始模型概率假设条件，随机从模型集 $\mathbb{M}$ 中按一定的模型概率产生该粒子的运动模型变量。

（2）对于“存活”粒子（即 $E_{k-1}^{(n)}=1$，$E_{k-1}^{(n)}=1$），首先通过前一时刻该粒子的模型变量 $r_{k-1}^{(n)}$ 及 Π_m 获取当前时刻的模型变量 $r_k^{(n)}$，然后再根据 $r_k^{(n)}$ 所对应的目标运动方程进行粒子状态预测。

步骤 3：粒子权值计算

采用似然比形式计算粒子的权重，归一化前的权重为

$$w_k^{(n)}=\begin{cases}\prod\limits_{i\in C_i(\boldsymbol{x}_k^{(n)})}\prod\limits_{j\in C_j(\boldsymbol{x}_k^{(n)})} l(z_k^{(i,j)}\mid \boldsymbol{x}_k^{(n)}) & E_k^{(n)}=1\\ 1 & E_k^{(n)}=0\end{cases}$$

式中：$C_i(\boldsymbol{x}_k^{(n)})=\{i_0-p,\cdots,i_0,i_0+p\}$，$i_0$ 表示最接近 $\boldsymbol{x}_k^{(n)}[1]$ 的整数，p 为受目标影响的单元个数，$C_j(\boldsymbol{x}_k^{(n)})$ 的含义与此类似。完成粒子权值计算后得到归一化的权值为 $\tilde{w}_k^{(n)}=w_k^{(n)}/\sum_{n=1}^{N}w_k^{(n)}$。

步骤 4：粒子重采样

重采样可选择性进行（即当有效粒子数小于某一特定值时才考虑重采样），也可强制进行，经重采样后得 $\{\boldsymbol{y}_k^{(n)},N^{-1}\}_{n=1}^{N}=\text{Resample}\left(\{\boldsymbol{y}_k^{(n)},w_k^{(n)}\}_{n=1}^{N}\right)$。

步骤 5：估计输出

首先估计目标的存在概率，即

$$\hat{P}_e=\frac{1}{N}\sum_{n=1}^{N}E_k^{(n)}$$

若该值大于预设门限 P_{th} 则认为目标存在，进而估计目标的状态

$$\hat{\boldsymbol{x}}_k=\sum_{n=1}^{N}(\boldsymbol{x}_k^{(n)}E_k^{(n)})/\sum_{n=1}^{N}E_k^{(n)}$$

同时，对于 $E_k^{(n)}=1$ 的粒子，分别统计模型指示变量 $r_k^{(n)}=1$、$r_k^{(n)}=2$ 及 $r_k^{(n)}=3$ 的数量 N_1、N_2 和 N_3，由此得到 k 时刻 3 种不同运动模型的概率估计

$$\hat{P}_{r_k=1}=N_1/\sum_{n=1}^{N}E_k^{(n)}\,,\quad \hat{P}_{r_k=2}=N_2/\sum_{n=1}^{N}E_k^{(n)}\,,\quad \hat{P}_{r_k=3}=N_3/\sum_{n=1}^{N}E_k^{(n)}$$

下面通过对机动目标的检测前跟踪仿真验证算法的有效性，实验中假设扫描周期 T=1s，过程噪声协方差矩阵为

$$\boldsymbol{Q}_k=\begin{bmatrix}\frac{q_1}{3}T^3 & \frac{q_1}{2}T^2 & 0 & 0 & 0\\ \frac{q_1}{2}T^2 & q_1T & 0 & 0 & 0\\ 0 & 0 & \frac{q_1}{3}T^3 & \frac{q_1}{2}T^2 & 0\\ 0 & 0 & \frac{q_1}{2}T^2 & q_1T & 0\\ 0 & 0 & 0 & 0 & q_2T\end{bmatrix}$$

式中：q_1=0.001；q_2=0.01。仿真持续时间为 45s，目标实际存在时间为 30s（其中 6s～15s 为匀速直线运动，16s～20s 为逆时针转弯机动，21s～30s 恢复为匀速直线运动，31s～35s 为顺时针转弯机动，目标在 36s 以后消失），目标出现时的初始位置、速度及信号强度分别为 (20,20)，(2,0) 和 I_0=20，其他参数设置见表 12.1。

表 12.1　MMPF-TbD 滤波算法参数设置

变量名	n_x,n_y	Δ_x,Δ_y	P_b,P_d	σ	Σ	P_{th}	γ	p	$I_{\min}$	$I_{\max}$	$v_{\max}$	α_m
参数值	50	1	0.05	1.63	0.7	0.6	0.05	3	15	25	2	1.08

进一步假定目标初始时刻的存在概率为 0.05，“新生”粒子的 3 种运动模型概率分别为 0.9、0.05 和 0.05，“存活”粒子的模型转换概率矩阵为

$$\Pi_m=\begin{bmatrix}0.90 & 0.05 & 0.05\\ 0.10 & 0.80 & 0.10\\ 0.10 & 0.10 & 0.80\end{bmatrix}$$

仿真中采用 N=60000 个粒子进行跟踪，所得结果如图 12.7～图 12.9 所示。

由图 12.7 中的结果可以看出，在仿真参数设置条件下（对应 12dB 信噪比），通过传感器观测得到的图像很难直接判断是否存在目标。但从图 12.8 中可以看到，通过多模粒子检测前跟踪算法，可成功检测到目标的出现和消失；且在本实例的参数设置条件下，对目标出现的检测无时延，而对目标消失的检测仅存在一个时刻的时延。图 12.9 的结果则表明，在正确检测的前提下，多模滤波算法可实现对目标的正确跟踪，且在机动发生阶段未出现明显的误差增大现象，说明了滤波算法的有效性。对多次实验进行统计分析，可得出类似的结论，在此不再赘述。

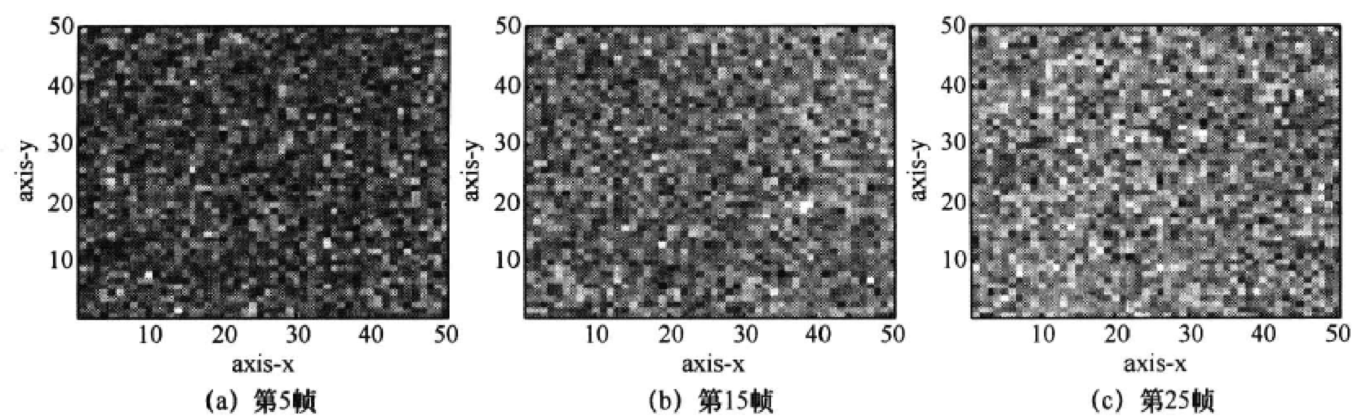


图 12.7　不同时刻的观测图像

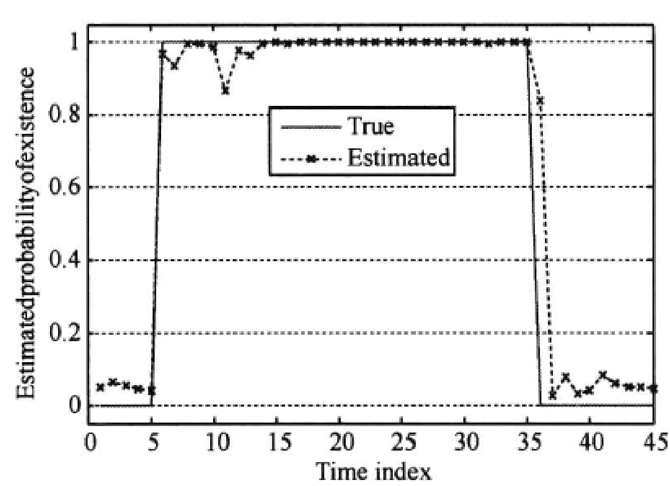


图 12.8　目标存在概率估计

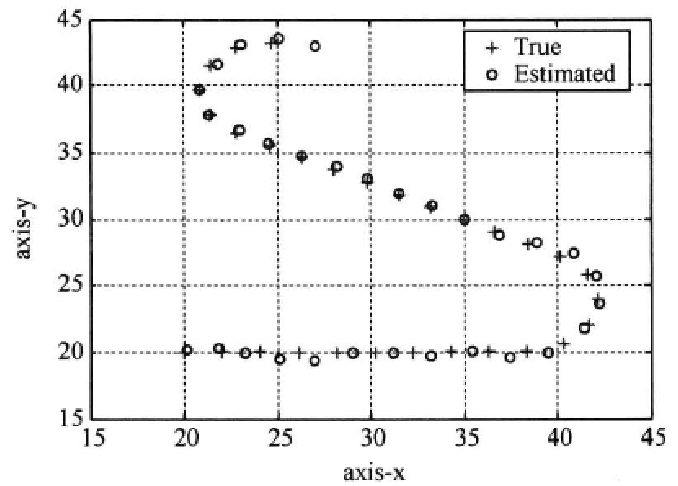


图 12.9　目标跟踪结果

12.5　本 章 小 结

本章主要讨论非线性滤波理论在多模条件下的扩展，首先给出了贝叶斯滤波框架下多模滤波问题解的一般表达式，在此基础上重点阐述基于混合高斯近似的多模滤波计算

问题，并以交互多模（IMM）为代表对其具体的实现方法进行了详细介绍。作为多模滤波的另一种重要的实现手段，本章还对基于粒子实现的多模滤波算法进行了详细分析，并通过实例验证粒子重采样过程在模型选择中的作用。

仿真结果表明，采用高斯混合近似和随机采样近似与多模型方法相结合，均可实现对机动目标的跟踪。相比较而言，高斯混合多模滤波方法运算复杂度较小，但应用范围不如粒子滤波方法广泛（如检测前跟踪应用通常只能通过粒子滤波的方法来实现）。因此，实际应用中需根据具体的问题选择合适的算法。一般而言，适合用高斯混合近似处理的问题，应尽量避免用复杂度较大的多模粒子滤波方法来处理。另外，本章介绍的几种滤波算法只是多模滤波方法中最典型、最重要的方法，关于这类算法的更多介绍和研究，可参阅文献[23]～文献[26]。

参 考 文 献

[1] 何友, 王国宏, 等. 多传感器信息融合及应用[M]. 北京: 电子工业出版社, 2007.

[2] X. R Li, Yunmin Z, Jie W, et al. Optimal linear estimation fusion—Part I: unified fusion rules[J]. IEEE Transactions on Information Theory, 2003, 49（9）: 2192–2208.

[3] 韩崇昭, 朱洪艳, 段战胜, 等. 多源信息融合[M]. 北京: 清华大学出版社, 2006.

[4] Lee-Ling O, Ben U, Matthew R, et al. Consistent methods for Decentralised Data Fusion using Particle Filters[C]. Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, 2006: 85–91.

[5] 熊伟, 何友, 张晶炜. 多传感器顺序粒子滤波算法[J]. 电子学报, 2005, 33（6）: 1116–1119.

[6] Waltz E, Llinas J. Multisensor Data Fusion [M]. London: Artech House, 1990.

[7] 段战胜, 韩崇昭. 相关量测噪声情况下多传感器集中式融合跟踪[J]. 系统工程与电子技术, 2005, 27（7）: 1160–1163.

[8] Jolliffe I T. Principal Component Analysis[M]. New York: Springer-Verlag, 2002.

[9] Tariq K, Pradeep R, Sarat C D. Particle-filter-based multisensor fusion for solving low-frequency electromagnetic NDE inverse problems[J]. IEEE Transactions on Instrumentation and Measurement, 2011, 60（6）: 2142–2153.

[10] Deok-Jin L. Nonlinear estimation and multiple sensors fusion using unscented information filtering[J]. IEEE Signal Processing Letters, 2008, 15: 861–864.

[11] Maskell S R, Weekes K R, Briers M. Distributed tracking of stealthy targets using particle filters[C]. Proceedings of IEE Seminar on Target Tracking: Algorithms and Applications, 2006: 13–20.

[12] Coates M. Distributed particle filters for sensor network[C]. Proceedings of the Third International Symposium on Information Processing in Sensor Networks, 2004: 99–107.

[13] Martin E L, David L H, James L. Handbook of Multisensor Data Fusion Theory and Practice（Second Edition）[M]. New York: CRC Press, 2008.

[14] 李善仓, 张德运, 杨振宇, 等. 无线传感器网络下的粒子滤波分布式目标跟踪算法[J]. 西安交通大学学报, 2007, 41（8）: 912–916.

[15] Ristic B, Arulampalam S, Gordon N. Beyond the Kalman: Particle filters for Tracking Applications[M]. London: Artech House, 2004.

[16] 杨小军. 基于粒子滤波的混合估计理论与应用[D]. 西安: 西北工业大学博士学位论文, 2006.

[17] Driessen H, Boers Y. Efficient particle filter for jump Markov nonlinear systems[J]. IEE Proceedings: Radar, Sonar and Navigation, 2005, 152（5）: 323–326.

[18] X. R Li, Jilkov V P. Survey of maneuvering target tracking-Part V: multiple-model method[J]. IEEE Transactions on Aerospace and Electronic Systems, 2005, 41（4）: 1255–1321.

[19] Mazor E, Averbuch A, Bar-shalom Y, et al. IMM methods in target tracking: a survey[J]. IEEE Transactions on Aerospace and Electronic Systems, 1998, 34（1）: 103–123.

[20] W R Wu, Cheng P P. A nonlinear IMM algorithm for maneuvering target tracking[J]. IEEE Transactions on Aerospace and Electronic Systems, 1994, 30（3）: 875–886.

[21] McGinnity S, Irwin G W. Multiple model bootstrap filter for maneuvering target tracking[J]. IEEE Transactions on Aerospace and Electronic Systems, 2000, 36（3）: 1006–1012.

[22] 刘先省, 胡振涛, 金勇, 等. 基于粒子优化的多模型粒子滤波算法[J]. 电子学报, 2010, 38（2）: 301–306.

[23] X. R Li, Bar-shalom Y. Performance prediction of the interacting multiple model algorithm[J]. IEEE Transactions on Aerospace and Electronic Systems, 1993, 29（3）: 755–771.

[24] 周云锋, 单甘霖, 张淼. 交互式多模型粒子滤波算法综述[J]. 军械工程学院学报, 2010, 22 （1）: 57–62.

[25] Ryan R P, Vesselin P J, X. R Li. A comparative study of multiple-model algorithms for maneuvering target tracking[C]. Proceedings of SPIE, 2005: 549–560.

[26] Kravaritis G, Mulgrew, B. Ground tracking using a variable structure multiple model particle filter with varying number of particles[C]. Proceedings of IEEE International Radar Conference, 2005: 837–841.

附　录

附录 A　矩阵求逆公式

A.1　分块矩阵求逆公式

若分块矩阵 $\boldsymbol{M}=\begin{pmatrix}\boldsymbol{A} & \boldsymbol{B}\\ \boldsymbol{C} & \boldsymbol{D}\end{pmatrix}$ 中 $\boldsymbol{A}$ 可逆，则有

$$\boldsymbol{M}^{-1}=\begin{pmatrix}\boldsymbol{A}^{-1}+\boldsymbol{A}^{-1}\boldsymbol{D}\boldsymbol{E}^{-1}\boldsymbol{C}\boldsymbol{A}^{-1} & -\boldsymbol{A}^{-1}\boldsymbol{D}\boldsymbol{E}^{-1}\\ -\boldsymbol{E}^{-1}\boldsymbol{C}\boldsymbol{A}^{-1} & \boldsymbol{E}^{-1}\end{pmatrix},\quad \boldsymbol{E}=\boldsymbol{D}-\boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B} \tag{A-1}$$

A.2　矩阵求逆引理

若矩阵 $\boldsymbol{A}$、$\boldsymbol{C}$ 均可逆，则有

$$(\boldsymbol{A}+\boldsymbol{B}\boldsymbol{C}\boldsymbol{D})^{-1}=\boldsymbol{A}^{-1}-\boldsymbol{A}^{-1}\boldsymbol{B}(\boldsymbol{D}\boldsymbol{A}^{-1}\boldsymbol{B}+\boldsymbol{C}^{-1})\boldsymbol{D}\boldsymbol{A}^{-1} \tag{A-2}$$

附录 B　式（1–78）证明

由 $\boldsymbol{S}_k=\boldsymbol{S}_{k|k-1}[\boldsymbol{I}-\boldsymbol{S}_{k|k-1}^{\mathrm{T}}\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{L}_k^{\mathrm{T}})^{-1}(\boldsymbol{L}_k^{\mathrm{T}}+\boldsymbol{R}_k^l)^{-1}\boldsymbol{H}\boldsymbol{S}_{k|k-1}]$ 可知

$$\begin{aligned}
\boldsymbol{S}_k\boldsymbol{S}_k^{\mathrm{T}}=&\boldsymbol{S}_{k|k-1}[\boldsymbol{I}-\boldsymbol{S}_{k|k-1}^{\mathrm{T}}\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{L}_k^{\mathrm{T}})^{-1}(\boldsymbol{L}_k+\boldsymbol{R}_k^l)^{-1}\boldsymbol{H}\boldsymbol{S}_{k|k-1}]\\
&[\boldsymbol{I}-\boldsymbol{S}_{k|k-1}^{\mathrm{T}}\boldsymbol{H}^{\mathrm{T}}[(\boldsymbol{L}_k+\boldsymbol{R}_k^l)^{\mathrm{T}}]^{-1}\boldsymbol{L}_k^{-1}\boldsymbol{H}\boldsymbol{S}_{k|k-1}]\boldsymbol{S}_{k|k-1}^{\mathrm{T}}\\
=&\boldsymbol{S}_{k|k-1}\{\boldsymbol{I}-\boldsymbol{S}_{k|k-1}^{\mathrm{T}}\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{L}_k^{\mathrm{T}})^{-1}(\boldsymbol{L}_k+\boldsymbol{R}_k^l)^{-1}\boldsymbol{H}\boldsymbol{S}_{k|k-1}-\boldsymbol{S}_{k|k-1}^{\mathrm{T}}\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{L}_k+\boldsymbol{R}_k^l)^{-1}\boldsymbol{L}_k^{-1}\boldsymbol{H}\boldsymbol{S}_{k|k-1}\\
&+\boldsymbol{S}_{k|k-1}^{\mathrm{T}}\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{L}_k^{\mathrm{T}})^{-1}(\boldsymbol{L}_k+\boldsymbol{R}_k^l)^{-1}\boldsymbol{H}\boldsymbol{S}_{k|k-1}\boldsymbol{S}_{k|k-1}^{\mathrm{T}}\boldsymbol{H}^{\mathrm{T}}[(\boldsymbol{L}_k+\boldsymbol{R}_k^l)^{\mathrm{T}}]^{-1}\boldsymbol{L}_k^{-1}\boldsymbol{H}\boldsymbol{S}_{k|k-1}\}\boldsymbol{S}_{k|k-1}^{\mathrm{T}}\\
=&\boldsymbol{P}_{k|k-1}-\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}}\{(\boldsymbol{L}_k^{\mathrm{T}})^{-1}(\boldsymbol{L}_k+\boldsymbol{R}_k^l)^{-1}+[(\boldsymbol{L}_k+\boldsymbol{R}_k^l)^{\mathrm{T}}]^{-1}\boldsymbol{L}_k^{-1}\\
&-(\boldsymbol{L}_k^{\mathrm{T}})^{-1}(\boldsymbol{L}_k+\boldsymbol{R}_k^l)^{-1}\boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}}[(\boldsymbol{L}_k+\boldsymbol{R}_k^l)^{\mathrm{T}}]^{-1}\boldsymbol{L}_k^{-1}\}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{P}_{k|k-1}
\end{aligned} \tag{B-1}$$

又根据 $\boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}}=\boldsymbol{L}_k\boldsymbol{L}_k^{\mathrm{T}}-\boldsymbol{R}_k$，得

$$\begin{aligned}
&(\boldsymbol{L}_k^{\mathrm{T}})^{-1}(\boldsymbol{L}_k+\boldsymbol{R}_k^l)^{-1}+[(\boldsymbol{L}_k+\boldsymbol{R}_k^l)^{\mathrm{T}}]^{-1}\boldsymbol{L}_k^{-1}-(\boldsymbol{L}_k^{\mathrm{T}})^{-1}(\boldsymbol{L}_k+\boldsymbol{R}_k^l)^{-1}\boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}}[(\boldsymbol{L}_k+\boldsymbol{R}_k^l)^{\mathrm{T}}]^{-1}\boldsymbol{L}_k^{-1}\\
&=(\boldsymbol{L}_k^{\mathrm{T}})^{-1}(\boldsymbol{L}_k+\boldsymbol{R}_k^l)^{-1}\{\boldsymbol{L}_k(\boldsymbol{L}_k+\boldsymbol{R}_k^l)^{\mathrm{T}}+(\boldsymbol{L}_k+\boldsymbol{R}_k^l)\boldsymbol{L}_k^{\mathrm{T}}-\boldsymbol{L}_k\boldsymbol{L}_k^{\mathrm{T}}+\boldsymbol{R}_k\}(\boldsymbol{L}_k+\boldsymbol{R}_k^l)^{-1}\boldsymbol{L}_k^{-1}\\
&=(\boldsymbol{L}_k^{\mathrm{T}})^{-1}(\boldsymbol{L}_k+\boldsymbol{R}_k^l)^{-1}\{\boldsymbol{L}_k\boldsymbol{L}_k^{\mathrm{T}}+\boldsymbol{L}_k(\boldsymbol{R}_k^l)^{\mathrm{T}}+\boldsymbol{R}_k^l\boldsymbol{L}_k^{\mathrm{T}}+\boldsymbol{R}_k\}[(\boldsymbol{L}_k+\boldsymbol{R}_k^l)^{\mathrm{T}}]^{-1}\boldsymbol{L}_k^{-1}\\
&=(\boldsymbol{L}_k^{\mathrm{T}})^{-1}(\boldsymbol{L}_k+\boldsymbol{R}_k^l)^{-1}(\boldsymbol{L}_k+\boldsymbol{R}_k^l)(\boldsymbol{L}_k+\boldsymbol{R}_k^l)^{\mathrm{T}}[(\boldsymbol{L}_k+\boldsymbol{R}_k^l)^{\mathrm{T}}]^{-1}\boldsymbol{L}_k^{-1}\\
&=(\boldsymbol{L}_k^{\mathrm{T}})^{-1}\boldsymbol{L}_k^{-1}=(\boldsymbol{L}_k\boldsymbol{L}_k^{\mathrm{T}})^{-1}
\end{aligned} \tag{B-2}$$

将式（B-2）代入式（B-1），得

$$\begin{aligned}\boldsymbol{S}_k\boldsymbol{S}_k^{\mathrm{T}} &= \boldsymbol{P}_{k|k-1} - \boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{L}_k\boldsymbol{L}_k^{\mathrm{T}})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{P}_{k|k-1} \\ &= \boldsymbol{P}_{k|k-1} - \boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R}_k)^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{P}_{k|k-1} \\ &= \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k\boldsymbol{H}\boldsymbol{P}_{k|k-1} = \boldsymbol{P}_k\end{aligned} \tag{B-3}$$

附录C　式（4–42）导出

对于滤波估计问题，测量更新的目的是在给定状态预测和观测量等信息的基础上寻求更优的状态估计。若将 $\hat{\boldsymbol{x}}_j$ 和 $\boldsymbol{y}$ 视为相互独立的多维高斯随机矢量的实现，则有

$$\hat{\boldsymbol{x}}_j \sim \mathcal{N}(\boldsymbol{x},\boldsymbol{P}_j), \quad \boldsymbol{y} \sim \mathcal{N}(h(\boldsymbol{x}),\boldsymbol{R}) \tag{C-1}$$

式中：$\boldsymbol{x}$ 和 $\hat{\boldsymbol{x}}_j$ 分别表示当前状态及其估值；$\boldsymbol{y}$ 表示实际观测值；$h(\boldsymbol{x})$ 为测量模型。

为表示方便，将两矢量合成为一增广的矢量形式，于是有

$$z = \begin{bmatrix}\boldsymbol{y} \\ \hat{\boldsymbol{x}}_j\end{bmatrix}, \quad g(\boldsymbol{x}) = \begin{bmatrix}h(\boldsymbol{x}) \\ \boldsymbol{x}\end{bmatrix} \tag{C-2}$$

由式（C-1）及前面的独立性假设条件，得

$$z \sim \mathcal{N}(g(\boldsymbol{x}),\boldsymbol{S}), \quad \boldsymbol{S} = \begin{bmatrix}\boldsymbol{R} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{P}_j\end{bmatrix} \tag{C-3}$$

于是，更新问题转化为在给定 z、$g(\boldsymbol{x})$ 和 $\boldsymbol{S}$ 条件下，寻求最优状态估计量 $\boldsymbol{x}^*$ 的问题。

若用自由变量 ξ 代替 $\boldsymbol{x}$，不难建立下列似然函数

$$\Lambda(\xi) = const \cdot \exp\left[-\frac{1}{2}(z-g(\xi))^{\mathrm{T}}\boldsymbol{S}^{-1}(z-g(\xi))\right] \tag{C-4}$$

$\boldsymbol{x}$ 的最大似然估计（最优估计）可表示为

$$\boldsymbol{x}^* = \arg\max[\Lambda(\xi)] \tag{C-5}$$

式（C-5）的等价表达式为

$$\boldsymbol{x}^* = \arg\min[q(\xi)] \tag{C-6}$$

式中

$$q(\xi) = \frac{1}{2}(z-g(\xi))^{\mathrm{T}}\boldsymbol{S}^{-1}(z-g(\xi)) \tag{C-7}$$

显然，要得到确切的 ξ^* 使之满足式（C-6）是不容易的，但若有

$$q(\xi_{j+1}) < q(\xi_j) \tag{C-8}$$

则可认为 $q(\xi_{j+1})$ 比 $q(\xi_j)$ 更接近最大似然面，等效地，ξ_{j+1} 就比 ξ_j 更接近最优解。

由式（C-7）和式（C-8）可得到如下不等式

$$\begin{aligned}&[\boldsymbol{y}-h(\xi_{j+1})]^{\mathrm{T}}\boldsymbol{R}^{-1}[\boldsymbol{y}-h(\xi_{j+1})] + (\hat{\boldsymbol{x}}_j-\xi_{j+1})^{\mathrm{T}}\boldsymbol{P}_j^{-1}(\hat{\boldsymbol{x}}_j-\xi_{j+1}) \\ &< [\boldsymbol{y}-h(\xi_j)]^{\mathrm{T}}\boldsymbol{R}^{-1}[\boldsymbol{y}-h(\xi_j)] + (\hat{\boldsymbol{x}}_j-\xi_j)^{\mathrm{T}}\boldsymbol{P}_j^{-1}(\hat{\boldsymbol{x}}_j-\xi_j)\end{aligned} \tag{C-9}$$

若序列 $\{\xi_j\}$ 由迭代产生，立即可得式（4-42）。

附录D　式（4–43）导出

对于正定矩阵序列 $\{\boldsymbol{P}_j\}$ ，其中 $\boldsymbol{P}_j=[p_{m,n}^{(j)}]\in\mathbb{R}^{L\times L}$ ，若对于 $j=1,2,\cdots$ 都有不等式 $\boldsymbol{P}_{j+1}<\boldsymbol{P}_j$ 成立，则可以证明矩阵序列 $\{\boldsymbol{P}_j\}$ 的极限存在。证明过程可分为两步：①矩阵序列对角元素 $p_{m,m}^{(j)}\ (m=1,2,\cdots,L)$ 的极限存在；②任意元素 $p_{m,n}^{(j)}(m,n=1,2,\cdots,L)$ 的极限存在。

证明①：

令 $\boldsymbol{e}_m=[0,\cdots,0,1,0,\cdots,0]^{\mathrm{T}}$ 为 L 维单位矢量，其中第 m 维的元素为 1 其他位置为 0。由已知条件 $\boldsymbol{P}_{j+1}<\boldsymbol{P}_j$ ，易得 $\boldsymbol{e}_m^{\mathrm{T}}\boldsymbol{P}_{j+1}\boldsymbol{e}_m<\boldsymbol{e}_m^{\mathrm{T}}\boldsymbol{P}_j\boldsymbol{e}_m$ ，即

$$p_{m,m}^{(j+1)}<p_{m,m}^{(j)}$$

由上式可知，$\{p_{m,m}^{(j)}\}$ 为单调递减序列，同时该序列是有界的，即 $0\leqslant p_{m,m}^{(j)}\leqslant p_{m,m}^{(1)}$ 。故该序列的极限存在，即有 $\lim\limits_{j\to\infty}p_{m,m}^{(j)}=c_{m,m}$ （$c_{m,m}$ 为一常量）。

证明②：

对任意两单位矢量 $\boldsymbol{e}_m$ 和 $\boldsymbol{e}_n$ ，若 $\boldsymbol{P}_{j+1}<\boldsymbol{P}_j$ ，则有

$$(\boldsymbol{e}_m^{\mathrm{T}}+\boldsymbol{e}_n^{\mathrm{T}})\boldsymbol{P}_{j+1}(\boldsymbol{e}_m+\boldsymbol{e}_n)<(\boldsymbol{e}_m^{\mathrm{T}}+\boldsymbol{e}_n^{\mathrm{T}})\boldsymbol{P}_j(\boldsymbol{e}_m+\boldsymbol{e}_n)$$

注意到 $\boldsymbol{P}_{j+1}$ 和 $\boldsymbol{P}_j$ 均为对称矩阵，于是有

$$p_{m,m}^{(j+1)}+p_{n,n}^{(j+1)}+2p_{m,n}^{(j+1)}<p_{m,m}^{(j)}+p_{n,n}^{(j)}+2p_{m,n}^{(j)}$$

利用①中的结论，显然有 $\lim\limits_{j\to\infty}p_{m,n}^{(j)}=c_{m,n}$ 。

综合①和②，说明矩阵序列 $\{\boldsymbol{P}_j\}$ 的极限存在，且 $\lim\limits_{j\to\infty}\boldsymbol{P}_j=\lim\limits_{j\to\infty}[p_{m,n}^{(j)}]=[c_{m,n}]=\boldsymbol{C}$ 。

附录E　式（8–5）导出

对于式（E-1）和式（E-2）所示的系统方程

$$\boldsymbol{x}_{k+1}=f(\boldsymbol{x}_k)+\boldsymbol{w}_k\qquad \boldsymbol{w}_k\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{Q}_k)\tag{E-1}$$

$$\boldsymbol{y}_k=\boldsymbol{H}_k\boldsymbol{x}_k+\boldsymbol{v}_k\qquad \boldsymbol{v}_k\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{R}_k)\tag{E-2}$$

式中：状态量和观测量的维数分别为 m 和 n。根据贝叶斯准则，有

$$p(\boldsymbol{x}_k\mid\boldsymbol{x}_{k-1},\boldsymbol{y}_k)=p(\boldsymbol{y}_k\mid\boldsymbol{x}_k)p(\boldsymbol{x}_k\mid\boldsymbol{x}_{k-1})/p(\boldsymbol{y}_k\mid\boldsymbol{x}_{k-1})\tag{E-3}$$

为表示方便，首先简记 $f(\boldsymbol{x}_{k-1})=f_k$ ，$\boldsymbol{x}_k^{\mathrm{T}}\boldsymbol{M}_k^{-1}\boldsymbol{x}_k=\|\boldsymbol{x}_k\|_{\boldsymbol{M}_k^{-1}}^2$ ，于是式（E-3）可写为

$$p(\boldsymbol{x}_k\mid\boldsymbol{x}_{k-1},\boldsymbol{y}_k)=\frac{[\det(\boldsymbol{H}_k\boldsymbol{Q}_k\boldsymbol{H}_k^{\mathrm{T}}+\boldsymbol{R}_k)]^{1/2}\exp\{-0.5[\|\boldsymbol{x}_k-f_k\|_{\boldsymbol{Q}_k^{-1}}^2+\|\boldsymbol{y}_k-\boldsymbol{H}_k\boldsymbol{x}_k\|_{\boldsymbol{R}_k^{-1}}^2]\}}{[(2\pi)^m\det(\boldsymbol{Q}_k)\det(\boldsymbol{R}_k)]^{1/2}\exp\{-0.5\|\boldsymbol{x}_k-\boldsymbol{H}_kf_k\|_{(\boldsymbol{H}_k\boldsymbol{Q}_k\boldsymbol{H}_k^{\mathrm{T}}+\boldsymbol{R}_k)^{-1}}^2\}}\tag{E-4}$$

注意到

$$\|\boldsymbol{x}_k-f_k\|_{\boldsymbol{Q}_k^{-1}}^2+\|\boldsymbol{y}_k-\boldsymbol{H}_k\boldsymbol{x}_k\|_{\boldsymbol{R}_k^{-1}}^2=\|\boldsymbol{x}_k-\boldsymbol{P}_k\boldsymbol{N}_k\|_{\boldsymbol{P}_k^{-1}}^2+\|f_k\|_{\boldsymbol{Q}_k^{-1}}^2+\|\boldsymbol{y}_k\|_{\boldsymbol{R}_k^{-1}}^2-\|\boldsymbol{N}_k\|_{\boldsymbol{P}_k}^2\tag{E-5}$$

式中：$\boldsymbol{P}_k^{-1}=\boldsymbol{Q}_k^{-1}+\boldsymbol{H}_k^{\mathrm{T}}\boldsymbol{R}_k^{-1}\boldsymbol{H}_k$ ；$\boldsymbol{N}_k=\boldsymbol{Q}_k^{-1}f_k+\boldsymbol{H}_k^{\mathrm{T}}\boldsymbol{R}_k^{-1}\boldsymbol{y}_k$ 。同时令 $\boldsymbol{D}_k=\boldsymbol{H}_k\boldsymbol{Q}_k\boldsymbol{H}_k^{\mathrm{T}}+\boldsymbol{R}_k$ ，并

对式（E-5）右边的后三项进行复杂的简化运算，最终可得

$$\begin{aligned}\|f_k\|_{Q_k^{-1}}^2+\|y_k\|_{R_k^{-1}}^2-\|N_k\|_{P_k}^2 &= y_k^{\mathrm{T}}D_k^{-1}y_k+f_k^{\mathrm{T}}H_k^{\mathrm{T}}D_k^{-1}H_kf_k \\ &\quad -f_k^{\mathrm{T}}H_k^{\mathrm{T}}D_k^{-1}y_k-y_k^{\mathrm{T}}D_k^{-1}H_kf_k \\ &=(y_k-H_kf_k)^{\mathrm{T}}D_k^{-1}(y_k-H_kf_k)\end{aligned} \tag{E-6}$$

进一步，式（E-4）变为

$$p(x_k\mid x_{k-1},y_k)=\frac{[\det(H_kQ_kH_k^{\mathrm{T}}+R_k)]^{1/2}\exp\{-0.5\|x_k-P_kN_k\|_{P_k^{-1}}^2\}}{[(2\pi)^m\det(Q_k)\det(R_k)]^{1/2}} \tag{E-7}$$

记 $K_k=P_kH_k^{\mathrm{T}}R_k^{-1}$，由块矩阵计算式，得

$$\begin{bmatrix}Q_k^{-1}+H_k^{\mathrm{T}}R_k^{-1}H_k & -H_k^{\mathrm{T}}R_k^{-1}\\ -R_k^{-1}H_k & R_k^{-1}\end{bmatrix}=\begin{bmatrix}I_m & -K_k\\ 0 & I_n\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}P_k^{-1} & 0\\ 0 & D_k^{-1}\end{bmatrix}\begin{bmatrix}I_m & -K_k\\ 0 & I_n\end{bmatrix}=A_1 \tag{E-8}$$

$$\begin{bmatrix}Q_k^{-1}+H_k^{\mathrm{T}}R_k^{-1}H_k & -H_k^{\mathrm{T}}R_k^{-1}\\ -R_k^{-1}H_k & R_k^{-1}\end{bmatrix}=\begin{bmatrix}I_m & 0\\ -H_k & I_n\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}Q_k^{-1} & 0\\ 0 & R_k^{-1}\end{bmatrix}\begin{bmatrix}I_m & 0\\ -H_k & I_n\end{bmatrix}=A_2 \tag{E-9}$$

根据行列式性质，有

$$\det(Q_k^{-1})\det(R_k^{-1})=\det(A_2)=\det(A_1)=\det(P_k^{-1})\det(D_k^{-1}) \tag{E-10}$$

于是

$$\begin{aligned}p(x_k\mid x_{k-1},y_k)&=[(2\pi)^m\det(P_k)]^{-1/2}\exp\{-0.5\|x_k-P_kN_k\|_{P_k^{-1}}^2\}\\ &=\mathcal{N}(x_k;P_kN_k,P_k)\end{aligned} \tag{E-11}$$

附录F　式（11-16）导出

$$\begin{aligned}p(x_k\mid y_{1:k}^{1:S})\propto p(y_k^{1:S}\mid x_k)p(x_k\mid y_{1:k-1}^{1:S})&=p(y_k^1y_k^2,\cdots y_k^s\mid x_k)p(x_k\mid y_{1:k-1}^{1:S})\\ &=p(x_k\mid y_{1:k-1}^{1:S})\prod_{s=1}^{S}p(y_k^s\mid x_k)\end{aligned} \tag{F-1}$$

对式（F-1）取对数，得到其等价算式为

$$\hat{x}_k^{\mathrm{MAP}}=\arg\min_{x_k}\{-\ln[p(x_k\mid y_{1:k-1}^{1:S})\prod_{s=1}^{S}p(y_k^s\mid x_k)]\} \tag{F-3}$$

容易证明，$p(x_k\mid y_{1:k-1}^{1:S})$ 和 $p(x_k\mid y_{1:k}^{1:S})$ 可以通过解析递推求得，且满足高斯分布，即

$$p(x_{k-1}\mid y_{1:k-1}^{1:S})=\mathcal{N}(x_{k-1};\hat{x}_{k-1},P_{k-1}) \tag{F-4}$$

$$\begin{aligned}p(x_k\mid y_{1:k-1})&=\mathcal{N}(x_k;\hat{x}_{k|k-1},P_{k|k-1})\\ &=\frac{1}{\sqrt{(2\pi)^m\,|P_{k|k-1}|}}\exp\left[-(x_k-\hat{x}_{k|k-1})^{\mathrm{T}}P_{k|k-1}^{-1}(x_k-\hat{x}_{k|k-1})\right]\end{aligned} \tag{F-5}$$

式中

$$\hat{\boldsymbol{x}}_{k|k-1} = \int \boldsymbol{x}_k p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1}) \mathrm{d}\boldsymbol{x}_k = \int \left[\int \boldsymbol{x}_k p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}) \mathrm{d}\boldsymbol{x}_k \right] p(\boldsymbol{x}_{k-1} \mid \boldsymbol{y}_{1:k-1}) \mathrm{d}\boldsymbol{x}_{k-1}$$
$$= \int \boldsymbol{F}\boldsymbol{x}_{k-1} p(\boldsymbol{x}_{k-1} \mid \boldsymbol{y}_{1:k-1}) \mathrm{d}\boldsymbol{x}_{k-1} = \boldsymbol{F}\hat{\boldsymbol{x}}_{k-1} \tag{F-6}$$

$$\begin{aligned} \boldsymbol{P}_{k|k-1} &= E\{[\boldsymbol{F}\boldsymbol{x}_{k-1} + \boldsymbol{w}_k - \boldsymbol{F}\hat{\boldsymbol{x}}_{k-1}][\boldsymbol{F}\boldsymbol{x}_{k-1} + \boldsymbol{w}_k - \boldsymbol{F}\hat{\boldsymbol{x}}_{k-1}]^{\mathrm{T}}\} \\ &= E\{[\boldsymbol{F}(\boldsymbol{x}_{k-1} - \hat{\boldsymbol{x}}_{k-1}) + \boldsymbol{w}_k][\boldsymbol{F}(\boldsymbol{x}_{k-1} - \hat{\boldsymbol{x}}_{k-1}) + \boldsymbol{w}_k]^{\mathrm{T}}\} \\ &= \boldsymbol{F}E[(\boldsymbol{x}_{k-1} - \hat{\boldsymbol{x}}_{k-1})(\boldsymbol{x}_{k-1} - \hat{\boldsymbol{x}}_{k-1})^{\mathrm{T}}]\boldsymbol{F}^{\mathrm{T}} + E[(\boldsymbol{w}_k \boldsymbol{w}_k^{\mathrm{T}})] \\ &= \boldsymbol{F}\boldsymbol{P}_{k-1}\boldsymbol{F}^{\mathrm{T}} + \boldsymbol{Q}_k \end{aligned} \tag{F-7}$$

又因为

$$p(\boldsymbol{y}_k \mid \boldsymbol{x}_k) = \prod_{s=1}^{S} \mathcal{N}(\boldsymbol{y}_k^s, \boldsymbol{H}^s \boldsymbol{x}_k, \boldsymbol{R}_k^s) \tag{F-8}$$

于是满足式（F-3）的 $\boldsymbol{x}_k$ 等价于

$$\begin{aligned} &\frac{\partial}{\partial \boldsymbol{x}_k} \sum_{s=1}^{S} [(\boldsymbol{y}_k^s - \boldsymbol{H}^s \boldsymbol{x}_k)^{\mathrm{T}} (\boldsymbol{R}_k^s)^{-1} (\boldsymbol{y}_k^s - \boldsymbol{H}^s \boldsymbol{x}_k)] + \\ &\frac{\partial}{\partial \boldsymbol{x}_k} [(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1})^{\mathrm{T}} \boldsymbol{P}_{k|k-1}^{-1} (\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1})] = 0 \end{aligned} \tag{F-9}$$

$$-\sum_{s=1}^{S} [(\boldsymbol{H}^s)^{\mathrm{T}} (\boldsymbol{R}_k^s)^{-1} (\boldsymbol{y}_k^s - \boldsymbol{H}^s \boldsymbol{x}_k)] + \boldsymbol{P}_{k|k-1}^{-1} (\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1}) = 0 \tag{F-10}$$

$$\begin{aligned} \boldsymbol{P}_{k|k-1}^{-1} (\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1}) &= \sum_{s=1}^{S} [(\boldsymbol{H}^s)^{\mathrm{T}} (\boldsymbol{R}_k^s)^{-1} (\boldsymbol{y}_k^s - \boldsymbol{H}^s \boldsymbol{x}_k)] \\ &= \sum_{s=1}^{S} \{(\boldsymbol{H}^s)^{\mathrm{T}} (\boldsymbol{R}_k^s)^{-1} [\boldsymbol{y}_k^s - \boldsymbol{H}^s (\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1} + \hat{\boldsymbol{x}}_{k|k-1})]\} \\ &= \sum_{s=1}^{S} [(\boldsymbol{H}^s)^{\mathrm{T}} (\boldsymbol{R}_k^s)^{-1} (\boldsymbol{y}_k^s - \boldsymbol{H}^s \hat{\boldsymbol{x}}_{k|k-1})] \\ &\quad - \{\sum_{s=1}^{S} [(\boldsymbol{H}^s)^{\mathrm{T}} (\boldsymbol{R}_k^s)^{-1} \boldsymbol{H}^s]\} (\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1}) \end{aligned} \tag{F-11}$$

$$\{\boldsymbol{P}_{k|k-1}^{-1} + \sum_{s=1}^{S} [(\boldsymbol{H}^s)^{\mathrm{T}} (\boldsymbol{R}_k^s)^{-1} \boldsymbol{H}^s]\} (\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1}) = \sum_{s=1}^{S} [(\boldsymbol{H}^s)^{\mathrm{T}} (\boldsymbol{R}_k^s)^{-1} (\boldsymbol{y}_k^s - \boldsymbol{H}^s \hat{\boldsymbol{x}}_{k|k-1})] \tag{F-12}$$

令 $\boldsymbol{P}_k = [\boldsymbol{P}_{k|k-1}^{-1} + \sum_{s=1}^{S} (\boldsymbol{H}_k^s)^{\mathrm{T}} (\boldsymbol{R}_k^s)^{-1} \boldsymbol{H}_k^s]^{-1}$ 乘以式（F-12）的两边可得 $\boldsymbol{x}_k$ 的融合估计为

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{P}_k \sum_{s=1}^{S} (\boldsymbol{H}^s)(\boldsymbol{R}_k^s)^{-1} (\boldsymbol{y}_k^s - \boldsymbol{H}^s \hat{\boldsymbol{x}}_{k|k-1}) \tag{F-13}$$

若定义

$$\boldsymbol{K}_k \triangleq (\boldsymbol{P}_{k|k-1}^{-1} + \boldsymbol{H}^{\mathrm{T}} \boldsymbol{R}^{-1} \boldsymbol{H})^{-1} \boldsymbol{H}^{\mathrm{T}} \boldsymbol{R}^{-1} = \boldsymbol{P}^{*} \boldsymbol{H}^{\mathrm{T}} \boldsymbol{R}^{-1} \tag{F-14}$$

对上式进行适当的变形，得

$$
\begin{aligned}
\boldsymbol{K}_k &= \boldsymbol{P}^* \boldsymbol{H}^{\mathrm{T}} \boldsymbol{R}^{-1} (\boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R})(\boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R})^{-1} \\
&= \boldsymbol{P}^* (\boldsymbol{H}^{\mathrm{T}} \boldsymbol{R}^{-1} \boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{H}^{\mathrm{T}} \boldsymbol{R}^{-1} \boldsymbol{R})(\boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R})^{-1} \\
&= \boldsymbol{P}^* (\boldsymbol{H}^{\mathrm{T}} \boldsymbol{R}^{-1} \boldsymbol{H} + \boldsymbol{P}_{k|k-1}^{-1}) \boldsymbol{P}_{k|k-1} \boldsymbol{H}^{\mathrm{T}} (\boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R})^{-1} \\
&= \boldsymbol{P}^* (\boldsymbol{P}^*)^{-1} \boldsymbol{P}_{k|k-1} \boldsymbol{H}^{\mathrm{T}} (\boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R})^{-1} \\
&= \boldsymbol{P}_{k|k-1} \boldsymbol{H}^{\mathrm{T}} (\boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R})^{-1}
\end{aligned} \tag{F-15}
$$

同理可得

$$
\begin{aligned}
\boldsymbol{P}_k &= E\{(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k)(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k)^{\mathrm{T}}\} \\
&= E\{[\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1} - \boldsymbol{K}_k(\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1})][\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1} - \boldsymbol{K}_k(\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1})]^{\mathrm{T}}\} \\
&= \boldsymbol{P}_{k|k-1}^{-1} - \boldsymbol{P}_{k|k-1}^{-1} \boldsymbol{H}^{\mathrm{T}} \boldsymbol{K}_k^{\mathrm{T}} - \boldsymbol{K}_k \boldsymbol{H} \boldsymbol{P}_{k|k-1}^{-1} + \boldsymbol{K}_k (\boldsymbol{H}\boldsymbol{P}_{k|k-1}^{-1}\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R}) \boldsymbol{K}_k^{\mathrm{T}}
\end{aligned} \tag{F-16}
$$

将式（F-15）中的 $\boldsymbol{K}_k$ 代入式（F-16），并采用矩阵求逆引理化简，得

$$
\begin{aligned}
\boldsymbol{P}_k &= \boldsymbol{P}_{k|k-1} - \boldsymbol{P}_{k|k-1} \boldsymbol{H}^{\mathrm{T}} (\boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R})^{-1} \boldsymbol{H}\boldsymbol{P}_{k|k-1} \\
&= (\boldsymbol{P}_{k|k-1}^{-1} + \boldsymbol{H}^{\mathrm{T}} \boldsymbol{R}^{-1} \boldsymbol{H})^{-1} = \boldsymbol{P}^*
\end{aligned} \tag{F-17}
$$